学号

密级

哈尔滨工程大学本科生毕业论文

基于需求预测的电商运输线路优化研究

——以X公司为例

|  |  |
| --- | --- |
| 学 院 名 称： | 经济管理学院 |
| 专 业 名 称： | 电子商务 |
| 学 生 姓 名： |  |
| 指 导 教 师： |  |

哈尔滨工程大学

2025年5月

学号

密级

基于需求预测的电商运输线路优化研究

——以X公司为例

Optimization of E-commerce Transportation Routes Based on Demand Forecasting: A Case Study of Company X

|  |  |
| --- | --- |
| 学生姓名： |  |
| 所在学院： | 经济管理学院 |
| 所在专业： | 电子商务 |
| 指导教师： |  |
| 职称： |  |
| 所在单位： | 哈尔滨工程大学 |
| 论文提交日期； | 2025年5月27日 |
| 论文答辩日期： | 2025年6月6日 |
| 学位授予单位： | 哈尔滨工程大学 |

摘 要

随着电子商务的快速发展和互联网技术的不断进步，电商物流网络的复杂性和动态性显著提升，如何精准预测运输需求并优化运输线路已成为提升企业供应链管理效率的关键。本文以X公司为研究对象，系统分析了其在物流网络运输线路优化中面临的需求预测不准、运输调度不灵活、网络结构瓶颈等突出问题。基于“数据驱动—智能优化”理论框架，构建了融合改进型粒子群优化算法（IPSO）与注意力机制长短时记忆网络（ALSTM）的运输需求预测模型，并结合网络流理论，提出了面向突发事件的紧急调运与运输线路优化方案。通过对2021—2022年X公司物流网络历史货量数据的实证分析，采用IPSO-ALSTM模型进行需求预测，并利用遗传算法、粒子群优化算法及其混合策略对网络流调度模型进行求解。

研究结果表明：（1）IPSO-ALSTM模型在运输需求预测的精度和鲁棒性方面均优于传统方法，能够有效捕捉季节性波动和突发事件影响；（2）混合优化算法在货流分配和网络调度中表现出更优的收敛速度和解的稳定性，显著提升了物流网络的应急响应能力和资源利用效率；（3）基于预测结果的动态调度策略能够有效缓解节点瓶颈和运输线路过载问题，保障物流网络在高峰期和异常情况下的平稳运行。

本文的主要贡献在于：第一，拓展了机器学习与运筹优化方法在电商物流管理中的集成应用，丰富了需求预测与运输优化的理论体系；第二，提出了面向实际业务约束的紧急调运模型，为电商企业提升供应链韧性和智能化管理水平提供了理论依据和实践参考。实践上，建议电商企业加强数据驱动的需求预测能力，完善物流网络的动态调度机制，并建立多元化的应急响应体系，以提升整体运营效率和客户服务水平。

关键词：电商物流；需求预测；IPSO-ALSTM；网络流

Abstract

With the rapid development of e-commerce and the continuous progress of Internet technology, the complexity and dynamics of the e-commerce logistics network have been significantly enhanced, and how to accurately predict the transportation demand and optimize the transportation routes has become the key to improve the efficiency of enterprise supply chain management. This paper takes Company X as the research object, and systematically analyzes the outstanding problems such as inaccurate demand prediction, inflexible transportation scheduling, and network structure bottleneck faced by the company in supply chain logistics management. Based on the theoretical framework of “data-driven-intelligent optimization”, we constructed a transportation demand forecasting model integrating improved particle swarm optimization algorithm (IPSO) and attention mechanism long-short-term memory network (ALSTM), and combined with the network flow theory, we put forward emergency dispatching and transportation route optimization scheme for unexpected events. transportation route optimization scheme. Through the empirical analysis of the historical cargo volume data of Company X's logistics network in 2021-2022, the IPSO-ALSTM model is used for demand forecasting, and the genetic algorithm, particle swarm optimization algorithm and its hybrid strategy are used to solve the network flow scheduling model.

The results show that (1) the IPSO-ALSTM model outperforms traditional methods in terms of both accuracy and robustness of transportation demand prediction, and is able to effectively capture seasonal fluctuations and the impact of unexpected events; (2) the hybrid optimization algorithm exhibits better convergence speed and stability of solution in cargo flow allocation and network scheduling, which significantly improves the emergency response capability and resource utilization efficiency of the logistics network; (3) the The dynamic scheduling strategy based on the prediction results can effectively alleviate the problems of node bottlenecks and overloaded transportation routes, and guarantee the smooth operation of the logistics network during peak periods and abnormal situations.

The main contributions of this paper are: firstly, it expands the integrated application of machine learning and operation research optimization methods in e-commerce logistics management, and enriches the theoretical system of demand forecasting and transportation optimization; secondly, it proposes an emergency dispatch model oriented to the actual business constraints, which provides the theoretical basis and practical reference for e-commerce enterprises to improve the resilience of the supply chain and the level of intelligent management. Practically, it is suggested that e-commerce enterprises strengthen their data-driven demand forecasting capability, improve the dynamic scheduling mechanism of the logistics network, and establish a diversified emergency response system in order to enhance the overall operational efficiency and customer service level.

**Keywords:** E-commerce Logistics; Demand Forecast; IPSO-ALSTM; Network Flow

目 录

[第1章 绪论 1](#_Toc196582885)

[1.1 研究背景 1](#_Toc196582886)

[1.2 研究目的和意义 2](#_Toc196582887)

[1.2.1研究目的 2](#_Toc196582888)

[1.2.2研究意义 2](#_Toc196582889)

[1.3 国内外研究现状及评述 3](#_Toc196582890)

[1.3.1 国外研究现状 3](#_Toc196582891)

[1.2.2 国内研究现状 4](#_Toc196582892)

[1.2.3 现有文献评述 6](#_Toc196582893)

[1.3 研究内容与方法 7](#_Toc196582894)

[1.3.1 研究内容 7](#_Toc196582895)

[1.3.2 研究方法 8](#_Toc196582896)

[1.4 论文创新之处 9](#_Toc196582897)

[1.5技术路线图 9](#_Toc196582898)

[第2章 概念界定与理论基础 11](#_Toc196582899)

[2.1 电商概念界定 11](#_Toc196582900)

[2.1.1 电商物流网络 11](#_Toc196582901)

[2.1.2 运输线路调度 11](#_Toc196582902)

[2.1.2 电商货量 11](#_Toc196582903)

[2.2 相关理论介绍 12](#_Toc196582904)

[2.2.1 IPSO模型理论 12](#_Toc196582905)

[2.2.2 ALSTM模型理论 12](#_Toc196582906)

[2.2.3 网络流理论 12](#_Toc196582907)

[2.3 本章小结 13](#_Toc196582908)

[第3章 X电商公司现状及问题分析 14](#_Toc196582909)

[3.1 X电商公司概况 14](#_Toc196582910)

[3.1.1 企业简介 14](#_Toc196582911)

[3.1.2 组织架构 14](#_Toc196582912)

[3.1.3 物流网络现状 15](#_Toc196582913)

[3.2 X电商公司物流网络问题 15](#_Toc196582914)

[3.2.1 运输线路优化问题概述 16](#_Toc196582915)

[3.2.2 成因分析 16](#_Toc196582916)

[3.3 本章小结 17](#_Toc196582917)

[第4章 基于时序数据的运输需求量预测 18](#_Toc196582918)

[4.1 历史数据采集与处理 18](#_Toc196582919)

[4.1.1 数据来源与数据集构建 18](#_Toc196582920)

[4.1.2 数据预处理与特征选择 19](#_Toc196582921)

[4.2 基于IPSO-ALSTM算法时间序列预测 22](#_Toc196582922)

[4.2.1 基于IPSO算法的超参数调优 22](#_Toc196582923)

[4.2.2 基于ALSTM的预测模型 24](#_Toc196582924)

[4.2.3 约束条件 26](#_Toc196582925)

[4.3 货量预测结果和分析 27](#_Toc196582926)

[4.3.1 IPSO-ALSTM模型损失值分析 28](#_Toc196582927)

[4.3.2 IPSO-ALSTM模型注意力权重分析 28](#_Toc196582928)

[4.3.3 IPSO-ALSTM模型预测结果 30](#_Toc196582929)

[4.4 本章小结 33](#_Toc196582930)

[第5章 基于网络流的紧急调运策略 34](#_Toc196582931)

[5.1 问题描述与建模背景 34](#_Toc196582932)

[5.1.1 问题背景 34](#_Toc196582933)

[5.1.2 模型构建目标 34](#_Toc196582934)

[5.2 数据分析与预处理 34](#_Toc196582935)

[5.2.1 数据概述 34](#_Toc196582936)

[5.2.2 数据预处理 35](#_Toc196582937)

[5.3 网络流优化与应急调度模型 37](#_Toc196582938)

[5.3.1 物流网络优化目标 37](#_Toc196582939)

[5.3.2 货流分配与调度 37](#_Toc196582940)

[5.3.3 应急响应机制 38](#_Toc196582941)

[5.4 算法设计与求解方法 38](#_Toc196582942)

[5.4.1 遗传算法（GA） 38](#_Toc196582943)

[5.4.2 粒子群优化（PSO） 39](#_Toc196582944)

[5.4.3 混合优化算法 39](#_Toc196582945)

[5.5 实验结果与分析 39](#_Toc196582946)

[5.5.1 收敛曲线分析 40](#_Toc196582947)

[5.5.2 算法性能比较 41](#_Toc196582948)

[5.6 本章小结 42](#_Toc196582949)

[第6章 对策及建议 43](#_Toc196582950)

[6.1 提升需求预测精度以应对货量波动 43](#_Toc196582951)

[6.2 构建动态调度系统以优化运输线路 43](#_Toc196582952)

[6.3 强化智能化技术和数据驱动管理 43](#_Toc196582953)

[结 论 44](#_Toc196582954)

[参考文献 46](#_Toc196582955)

[攻读学士学位期间发表的论文和取得的科研成果 52](#_Toc196582956)

[致 谢 53](#_Toc196582957)

[附 录 55](#_Toc196582958)

1. 绪论

1.1 研究背景

近年来，随着互联网和电子商务的迅猛发展，电商市场已经成为全球经济的一个重要组成部分。据统计，2024年全球电商市场规模已经超过4.8万亿美元[1]。在如此庞大的市场背景下，电商企业面临的物流问题越来越复杂和关键。快速、准确、高效的物流服务已成为电商企业竞争力的重要因素[2]。然而，物流网络的建设和优化面临许多挑战，包括如何预测货物需求、如何在突发情况下重新规划运输线路以及如何评估和优化整个物流网络的运作。

在这样的大背景下，电商物流的高效运作和管理面临着诸多挑战与机遇，其中电商物流的调度及结构最优化问题引起了学术界和业界的广泛关注。如何合理安排物流资源、优化物流网络结构以实现成本控制、效率提升以及服务质量保障，成为了各界学者深入探讨的热点方向。特别是电商物流网络具有高度的复杂性与动态性，它由众多物流场地以及它们之间相互交织的运输线路所构成[3]。在实际运营过程中，由于电商平台大促活动的周期性开展，如黑色星期五、双11、618等购物狂欢节，会导致用户下单量在特定时间段内出现显著的峰值波动，这对物流网络的承载能力和调配能力提出了极高要求。同时，不可忽视的是，诸如自然灾害、暴雨封路、疫情封控等不可抗力因素也时有发生。这些突发事件可能致使物流场地出现暂时性的运营滞涨甚至永久性的停运状况，进而直接影响到各条运输线路上流通的包裹数量，使其偏离正常的运行轨道，产生异常波动情况，严重阻碍整个电商物流网络的顺畅运行，给电商企业带来巨大的经济损失和声誉损害。

X电商公司是一家供应链技术驱动的电子商务公司，其营业模式主要为POP商家入驻和寄售仓储一体化平台管理。X公司作为中国智能物流网络运输线路优化先行者，其物流运输线路优化面临着更为严峻的挑战。一方面，由于X公司的物流网络需要协调多方资源，包括商家的库存管理、仓储中心的包裹处理以及最后一公里配送。这种多方协作的模式虽然为平台提供了灵活性和多样化的商品供给，但也显著增加了物流调度的复杂性。另一方面，X公司现有技术的实施可能面临数据孤岛、算法效率瓶颈以及高昂的初期投入成本。实时调度系统需要整合来自商家、仓库和配送端的海量数据，但如果数据接口不统一或更新不及时，可能导致调度决策失误。一旦系统无法快速适应异常情况，可能导致包裹积压、配送延误，甚至引发客户投诉和品牌声誉受损。

针对这些挑战，本研究将利用X公司的历史数据，建立基于时序数据的运输需求量预测模型和基于网络流的物流网络优化模型，来解决X电商公司在物流网络运输线路优化中遇到的问题。通过对时间序列数据的预测，可以准确预测未来的运输需求量；通过优化模型，可以在面对物流节点异常情况时，快速找到最优的运输路径，以保障物流网络的高效运行。

1.2 研究目的和意义

1.2.1研究目的

文章旨在结合IPSO和ALSTM算法对电商物流网络进行需求预测，并结合仓储分拣能力和线路运输能力，基于网络流设计出紧急分拨调运方案。核心目标是提升电商物流网络的运营效率，减少因突发事件（如节假日促销、仓储关停等）引起的运输中断或延误，确保货物的高效流转。通过这些模型，希望能够为我国电商企业在物流网络管理策略上提供新的视角和方法，从而提高供应链的响应速度和抗网络能力，优化物流管理效率。具体而言，本研究的目标包括：

（1）建立高精度的需求预测模型：采用改进粒子群优化算法（IPSO）与注意力机制长短期记忆网络（ALSTM）相结合的方法，精准预测电商物流网络的运输需求，为资源调度和运输规划提供数据支撑。

（2）设计和实现紧急分拨调运方案：在预测到突发事件或仓储关停风险时，基于网络流优化模型，快速调整运输路径和资源分配，保障物流网络的连续性和稳定性，进一步通过优化物流网络，提高运输效率，减少运输成本。

1.2.2研究意义

在电商物流网络优化领域的创新性贡献和实际应用价值是研究的重要维度，电商物流网络的动态性和复杂性（如订单峰值、突发中断）使得传统静态优化方法难以适用。因此，围绕真实的业务数据来研究物流优化问题有相当重要的理论意义和现实意义。

（1）理论意义

首先，本研究通过结合IPSO和ALSTM算法进行需求预测，丰富了电商物流管理理论的应用，为供应链管理理论在电商领域的应用提供了新的视角和方法。同时，本研究将网络流优化算法应用于电商物流线路优化，拓展了运输优化理论在实际业务中的应用，为物流管理理论体系的发展提供了新的思路。

其次，本文对电商物流网络中的需求预测和线路优化进行了深入研究，有助于填补在这一领域中现有研究的空白。通过将机器学习和优化算法应用于实际问题的解决，不仅拓展了电商物流管理的研究对象和场景，还为企业提供了更加科学和有效的管理策略，有助于提高企业的运营效率和供应链履约能力。

最后，本研究还对智能预测模型和网络流优化算法的结合进行了深入探讨，拓宽了这些技术在电商物流管理中的应用领域。这种结合不仅可以提高需求预测的准确性，还可以优化运输线路，降低运输成本，提高货物的流转效率。

（2）实践意义

从电商行业层面看，本研究强调了对市场需求和运输路径的高效管理和优化。通过提高需求预测的准确性和优化运输线路，可以有效降低由于信息不对称和突发事件引起的运输延误和额外成本，提高整个电商行业的供应链效率。研究提出的预测模型和动态调度模型具有普适性，可为我国电商物流行业提供借鉴，推动行业向智能化、韧性化发展，特别是在供应链管理效率提升和可持续发展方面具有参考价值。

从X电商公司的角度看，本研究通过精准的需求预测和快速的运输路径优化，针对X公司的具体需求，提供了一个全面、科学的物流管理优化方案。通过优化库存管理、需求预测和运输线路，X公司可以显著降低物流成本，降低因运营中断导致的经济损失和声誉风险，优化资源配置效率，提高库存周转率，增强企业的竞争力和可持续发展能力。

1.3 国内外研究现状及评述

1.3.1 国外研究现状

（1）需求预测的相关研究

随着亚马逊、eBay等电商平台的兴起而快速发展，国外对电商物流需求预测的研究起步于20世纪末，较早于我国，研究主要集中在传统统计方法、机器学习技术、多因素分析和不确定性管理等多个方面。早期研究多聚焦于传统统计方法的改进与应用。Mark Moon（1998）将需求预测从单纯的统计分析提升至战略决策层面，提出通过过程指标和结果指标评估预测绩效，为企业决策提供支持[4]。Zanfei等（2019）提出了一种图卷积递归神经网络（GCRNN），用于预测供水系统的区域性需水量时间序列，验证了其在捕捉空间依赖关系方面的优势[5]。Chaturvedi等（2020）对比分析了SARIMA、LSTM RNN和Fb Prophet模型在月度需求预测中的表现，指出Fb Prophet在处理长期趋势和短期季节性波动时更具优势[6]。

随着机器学习技术的发展，国外研究逐渐转向深度学习和多源数据融合。Milan等（2022）提出了T-LSTM-Enc与时态LSTM级联模型，用于充电需求预测，显著提升了模型对动态变化的适应性[7]。Tang等（2022）结合智能仓库技术和轮盘赌遗传算法，提出了一种数字双框架，用于周期性行业需求预测，并在库存优化中展现了应用价值[8]。Jaouhari等（2023）开发了一种基于物联网的库存管理系统（IMS），通过多元线性回归（MLR）和遗传算法（GA）结合，提升了客户需求预测的准确性，为工业4.0提供了智能库存解决方案[9]。Zhang等（2024）基于多源数据融合，构建了适用于电商物流的高峰期预测模型，结合不确定性风险管理，为物流资源配置提供了支持[10]。

（2）运输线路优化的相关研究

网络线路优化相关研究已形成较为成熟的体系，覆盖传统优化算法、动态调度技术、智能化方法和突发事件应对等多个领域。早期研究主要集中于传统优化算法的应用。Ford和Fulkerson（1962）提出的网络流理论为线路优化奠定了基础，其最大流和最小费用流算法被广泛应用于物流调度[11]。Ahuja等（1993）进一步发展了网络流算法，提出用于大规模运输问题的改进型Dinic算法，显著提升了求解效率[12]。Barnhart等（2001）基于车辆路径问题（VRP），结合混合整数规划（MIP），优化了多目标物流网络，实现了运输成本和时效的平衡[13]。

随着电商物流动态性需求的增加，动态调度和智能化技术成为研究热点Laporte等（2010）将遗传算法（GA）与模拟退火（SA）结合，开发了一种混合优化算法，用于解决多车辆配送线路问题，验证了其在减少配送延迟中的有效性[14]。Crainic等（2018）引入强化学习（RL），提出了一种动态调度框架，通过实时学习物流节点状态，优化运输路径，提升了网络的抗压能力[15]。Qiu（2016）基于鲁棒优化理论，提出了一种应对自然灾害的物流网络重配模型，通过蒙特卡洛模拟优化备用线路选择，保障物流连续性[16]。Nisha等（2023）结合物联网和实时数据，构建了一个动态调整框架，减少了配送中断[17]。

1.2.2 国内研究现状

（1）需求预测的相关研究

国内对电商物流需求预测的研究起步于2010年左右，虽然晚于国外，但近年来在理论构建和应用实践方面取得了显著进展，研究主要集中在传统预测方法改进、机器学习技术应用、场景化预测模型以及多源数据融合等多个方面。早起多集中于传统预测方法的改进与应用。马韶光（2015）基于ARIMA模型，根据大量的历史社会消费品的数据采用ARIMA模型分析规律，并且对未来趋势做预测[18]。李梦婉等（2017）则通过引入灰色预测模型（GM(1,1)），给出了基于求解优化和多项式拟合优化相结合的改进灰色等维动态预测方法，指出其改进后的适用性[19]。

近年来，机器学习技术在国内需求预测研究中的应用逐渐成为主流。徐甜甜（2018）提出了一种基于长短期记忆网络（LSTM）的需求预测模型，对股票涨跌进行设计建模，考虑了不同的组合以提高精度为诉求，该模型在股市预测中已经能取得了一定的实证效果[20]。高学金等（2022）结合注意力机制（Attention Mechanism）和LSTM，提出了ALSTM模型，对划分后的各个阶段分别建立注意力长短期记忆(LSTM)集成质量预测模型[21]。此外，刘拥民等（2023）将XGBoost算法与时间序列特征相结合，基于分解集成框架以及相关性去噪，实现了更准确的预测[22]。

与此同时，国内学者注重结合中国电商市场的实际场景（例如双11、618促销），开发适应性更强的预测模型。常恬君等（2021）基于多源数据（订单量、搜索指数、社交媒体热度），提出了适用于节假日促销的混合预测模型，综合运用随机森林和Prophet算法，提高了预测精度[23]。李长亮（2022）则聚焦于最后一公里配送需求，提出了一种基于时空特征的预测模型（ST-GCN），通过分析城市区域的订单分布和交通数据，优化配送资源配置[24]。

（2）运输线路优化的相关研究

国内对电商物流网络线路优化的研究近年来取得了长足进展，研究主要集中在线路规划算法、动态调度技术、突发事件应对以及智能化优化技术等多个方面。蒋璐璐（2016）基于车辆路径问题（VRP），提出了一种改进的蚁群算法（ACO），用于优化电商物流的多车辆配送线路，研究表明该方法在降低运输成本和缩短配送时间方面具有显著效果[25]。陈彦等（2018）结合混合整数规划（MIP），构建了一个多目标优化模型，综合考虑运输成本、配送时效和碳排放，优化区域配送网络[26]。然而，传统优化方法在面对动态需求和突发事件时，往往缺乏实时性和适应性。

其次，动态调度技术成为应对电商物流网络复杂性的重要研究方向。国内学者针对订单高峰期（如双11、618）和突发事件（如疫情封控），开发了多种动态调度算法以优化线路。例如，王超等（2020）提出了一种基于遗传算法（GA）的动态线路优化方法，通过实时分析订单数据和交通状况，动态调整配送路径[27]。宋涛（2021）则将强化学习（RL）引入线路优化，使用基于MAXQ算法的分层强化学习进行路径规划，从而解决了细粒度栅格单元带来的维度灾难问题，显著提高了网络的抗压能力[28]。此外，冯旭旸（2023）结合图神经网络（GNN），开发了一种异构控制物流网络的动态优化模型，通过捕捉节点间的时空依赖关系，可以优化最后一公里配送线路[29]。

突发事件下的网络线路优化是国内研究的重点方向。丁琳（2022）基于网络流理论，提出了一种水文分拨调运模型，在预测到突发状况时，快速重新规划运输路径，研究表明该模型有效减少了配送中断[30]。赵栋等（2023）则结合蒙特卡洛模拟和鲁棒优化，构建了一个适应不确定性环境的线路优化框架，通过模拟不同高速公路突发情景，优化备用线路的选择[31]。慧鹏等（2024）进一步提出了一种多目标鲁棒优化模型，综合考虑成本、时效和风险，优化了物流网络调度策略[32]。

1.2.3 现有文献评述

得益于国内外学者的辛勤耕耘，需求预测及库存管理的理论、方法已经越来越丰富，并在各行各业得以实践验证。但是，国内外关于电商物流方面的较多研究侧重于网络要素的识别，通过预测和优化方法实现网络控制领域研究相对较少，同时定性分析研究的文献相对较多，定量分析研究的文献相对较少。在CNKI中搜索“电商物流线路”检索词，共得到827条检索结果，其中博士学位论文共有14条，但对电商物流体系中节点和线路的控制的研究相对比较缺乏。

国外需求预测和网络线路优化研究起步早，形成了从传统统计到机器学习的完整体系，注重多源数据融合和不确定性管理，展现了较高的理论深度和应用价值。但其研究多集中于大型平台，针对中小型企业的预测方法较少，且模型复杂性可能导致实际部署成本较高，且对成本与服务质量的平衡关注较少。未来可探索轻量化算法和多目标优化方法，以提升普适性和经济性。

国内在电商物流需求预测领域的研究起步较晚，但发展迅速，形成了从传统方法改进到机器学习技术应用的完整研究体系。针对双11等促销场景的定制化模型体现了国内研究的实践导向，而多源数据融合和智能化预测则反映了与国际研究趋势的接轨。然而，国内研究仍存在不足：一是数据获取和标准化程度较低，限制了模型通用性；二是针对中小型企业的预测模型较少；三是智能化预测系统的实际部署成本较高。

综上所述，现有的研究多从单一角度，目前针对电子商务物流行业特别是线路运输量需求预测的文献比较少。此外，而尽管有不少学者将物流管理模式下的需求预测与物流网络管理集成研究，但在电子商务行业对需求预测与网络管理集成研究的文献还非常少，以往的研究较少考虑因区域分异特征导致的区域需求差异化配额问题和实际物流网络突发问题。同时，由于电商企业对物流时效的要求，传统的反应性物流配送模式已经无法满足现实需求。电商物流调拨优化问题需要综合考虑库存管理、订单处理、物流路径规划等多个方面，以实现更高效、更灵活的物流服务。所以本文将提出针对性的方法运用到电商物流调拨模式。

1.3 研究内容与方法

1.3.1 研究内容

研究旨在通过机器学习算法预测电商物流需求，并结合仓储分拣能力和线路运输能力，设计物流紧急调运方案并提出优化思路。最终目标是为电子商务企业提供理论和实践上的参考与借鉴，以增强其物流管理的效率和效果。

研究内容包括机器学习在需求预测中的应用和多目标线性规划模型构建。利用机器学习算法，如深度学习、集成学习等，对电商仓储模式和物流网络历史数据进行系统分析，构建有效的货运量预测模型。通过提高需求预测的精度，助力电商企业实现更精准的库存管理和降低库存成本。同时立足多目标线性规划理论，构建一个能够同时考虑多个目标（如成本最小化、服务水平最大化、网络最小化等）的结合物资紧急调运优化模型，考虑实际问题中的各种约束条件，建立数学模型以制定合理的调度方案。

第1章，绪论。对研究的背景、目的与意义进行阐述，包括电商的发展现状与挑战、物流网络管理的重要性以及机器学习在物流网络管理中的应用。综述国内外在电商物流网络管理研究和机器学习应用方面的研究现状。介绍研究内容与方法，明确研究的具体方向和所采用的科研方法。总结研究的创新点与不足，指出本研究的新颖性和局限性。

第2章，相关概念与理论基础。介绍电子商务模式、物流网络管理理论以及机器学习模型与算法等基础概念。为后续章节的深入研究提供理论和概念上的支持。强调理论基础对研究的重要性。

第3章，X电商公司现状及问题分析。本章对X电商公司的物流网络运输线路优化现状进行了深入分析，揭示了公司面临的具体问题和挑战，并分析了这些问题产生的原因，为后续的模型设计和优化提供了依据。

第4章，基于时序数据的运输需求量预测。介绍研究背景和研究问题。采集与处理历史数据，构建数据集并进行预处理。构建预测模型，选择与比较不同的模型，并进行训练与调优。展示实验结果，评估预测性能，并分析预测结果的应用。总结预测研究的主要发现。

第5章，基于网络流的紧急调运策略。分析电商物流管理决策周期和潜在关停网络的影响。构建基于多目标优化的紧急调运模型，明确模型目标与约束条件。选择特定算法进行模型求解，并进行适应性改进和实现优化。展示实验设计与模拟结果，评价模型效果并进行分析。总结紧急调运优化研究的关键点。

第6章，对策建议。本章总结了全文的研究成果，并讨论了如何将这些成果应用于实际业务中。包括如何优化物流管理、如何应对突发事件，以及如何提升供应链履约能力等具体对策建议。

1.3.2 研究方法

（1）文献研究法

全面搜集国内外电商物流领域相关文献资料，涵盖电商物流风险管理、机器学习在物流中的应用、仓储管理、运输调度等多个方面的学术论文、行业报告、专业书籍等。深入剖析这些文献中关于电商物流网络构建、物流需求预测模型、紧急调运策略以及相关优化算法等内容的研究成果与进展。系统梳理不同学者和研究机构在电商物流各环节所提出的理论框架、方法模型以及实践案例，从中提取出可用于本研究的关键思路、技术手段以及经验教训，为构建电商物流需求预测模型和紧急调运模型提供坚实的理论基础与方法借鉴，明确研究在现有学术体系中的位置与创新方向，避免重复前人的研究工作，同时确保研究的科学性与前瞻性。

（2）实证分析法

以X电商公司的实际运营数据为基础，通过对2021—2022年物流网络历史货量数据的实证分析，验证所提出的需求预测模型和网络优化方案的有效性。具体而言，首先对历史数据进行清洗、预处理和特征提取，构建适用于时序预测和网络优化的数据集；然后，利用IPSO-ALSTM模型对运输需求量进行预测，并通过遗传算法（GA）、粒子群优化算法（PSO）及其混合策略求解网络流调度模型；最后，通过对比实验和性能评估，分析模型在不同场景下的预测精度、鲁棒性以及优化效果。实证分析法不仅能够验证理论模型的可行性，还能为X电商公司的物流管理实践提供数据驱动的决策支持，确保研究成果具有较高的应用价值。

（3）数学建模法

基于机器学习和运筹优化理论，构建需求预测模型和物流网络优化模型，以实现运输需求的精准预测和资源的高效调度。在需求预测方面，融合改进型粒子群优化算法（IPSO）与注意力机制长短时记忆网络（ALSTM），构建IPSO-ALSTM预测模型，通过数学公式定义损失函数、约束条件以及超参数调优过程，确保模型能够捕捉复杂的时序特征和突发事件影响。在网络优化方面，基于网络流理论，构建多目标优化模型，明确货量守恒、运输能力、非负性等约束条件，并通过目标函数最小化路径负荷偏差和货流调整幅度，实现物流网络的动态调度和应急响应。数学建模法为研究提供了严谨的理论框架和定量分析工具，确保模型设计和求解过程的科学性与可操作性。

1.4 论文创新之处

本文不仅丰富了电商物流管理的理论体系，也为实际的物流运营提供了切实可行的优化方案。具有以下几个创新点：

（1）结合IPSO和ALSTM算法进行需求预测：本文创新地将IPSO优化算法和ALSTM预测算法进行融合，以提高预测的准确性。这种方法能够更好地捕捉市场需求的季节性变化和突发事件带来的影响，为后续的物流调度和资源配置提供了更为精确的依据。

（2）应用网络流优化算法进行运输线路优化：在面对突发事件（如仓储关停）时，本文采用网络流优化算法，设计了一套灵活的运输线路优化方案。这种方法可以快速响应突发情况，有效降低因突发事件引起的运输延误和额外成本，保障货物的及时递送。

1.5技术路线图

本研究以X电商公司物流网络运输为背景，聚焦于电商物流网络中包裹应急调运与结构优化问题，旨在通过预测与优化技术提升物流网络的效率和抗风险能力。采用文献分析法和机器学习算法构建线路货量预测模型，预测未来货量分布，并基于网络流理论和优化算法设计物流网络的动态调整与优化方案。本文研究的技术路线如图1.1所示。首先，在文献收集、整理与分析阶段，通过对国内外电商物流网络、运输线路调度、应急保障与网络鲁棒性等领域的最新成果进行梳理，明晰了包裹量波动对运营成本与服务水平的深远影响，并确立了以“预测-优化”协同为核心的研究范式。在模型设计与实证分析阶段，本文首先利用IPSO-ALSTM算法对历史时序数据进行特征提取与多步预测，获得各线路未来时段的包裹流量分布；继而基于预测结果构建动态网络流优化模型，引入应急调运策略与约束，设计针对突发需求的弹性调整方案。通过大规模仿真实验，对比基础调度、规则驱动与算法驱动三种方案的成本-时效-鲁棒性表现，验证了所提框架的有效性与算法的优越性。最后，在解决问题、得出结论环节，论文总结了研究发现，提出适用于电商企业的运输网络分级分区优化策略与应急资源储备建议。



图1.1 研究框架

第2章 概念界定与理论基础

2.1 电商概念界定

2.1.1 电商物流网络

电商物流网络是指在电子商务环境下，通过信息技术和物流技术相结合，实现商品从供应商到消费者手中的全过程管理[33]。它包括了订单处理、仓储管理、运输管理、配送管理等多个环节，旨在通过优化这些环节，提高供应链的效率和响应速度，降低物流成本，提升供应链履约能力。电商物流网络的最终目标是提高供应链履约能力，确保商品能够按时、按量、按质地送达消费者手中，满足消费者的需求[34]。

2.1.2 运输线路调度

物流运输线路调度是指在物流系统中，通过科学的方法和技术手段，对运输线路和车辆的安排进行规划、优化和动态调整，以实现货物从起点到终点的有效运输[35]。其核心目标是提高运输效率、降低运营成本、缩短配送时间，同时满足客户需求和应对各种约束条件，例如交通状况、车辆容量、时间窗口。物流运输线路调度涉及线路规划、车辆调度、动态调整等关键要素。针对动态调整，指的是在运输过程中，针对突发事件，实时调整线路和调度方案，以保障运输的连续性和效率。常借助传统优化算法（如网络流算法、混合整数规划）、智能化方法（如遗传算法、强化学习、图神经网络）以及实时数据（如物联网、GPS）来实现调度优化。

2.1.2 电商货量

电商货量是指在电商平台上，消费者通过在线购物产生的订单数量和货物体积[36]。这电商物流网络管理的核心对象，直接影响到物流网络的设计、运营和优化。在本研究中，通过对电商货量的深入分析和预测，结合X电商公司的实际情况，旨在建立一个高效的需求预测模型，并基于此优化运输线路，提高供应链履约能力，降低物流成本，提高运营效率。其特点主体现在波动性、时效性和数据驱动三方面。波动性主要来自促销活动、节假日效应和平台流量分层等外生冲击，时效性体现在消费者对“次日达”“当日达”服务的期望不断提高，货量分布随时间窗口迅速迁移，数据驱动属性则为深度学习模型提供持续迭代的样本流。

2.2 相关理论介绍

2.2.1 IPSO模型理论

粒子群优化（PSO）算法是一种基于群体智能的启发式全局优化方法，其灵感来源于鸟群觅食行为，已在众多高维非凸优化问题中得到广泛应用[37]。然而，在实际的深度学习超参数调优问题中，由于参数空间复杂且搜索空间维数较高，传统PSO容易陷入局部最优解且收敛速度较慢。为此，本文在传统PSO基础上进行了改进，提出了改进型PSO（IPSO）算法，用于优化ALSTM模型的超参数组合，从而提高预测模型的整体性能。

IPSO算法的基本流程包括：初始化粒子群、定义适应度函数、引入动态惯性权重与动态加速因子、以及粒子速度与位置的更新。具体而言，首先在初始化阶段，为每个粒子设定初始位置和初始速度，其位置向量代表当前候选超参数组合，包括ALSTM单元数、Dropout率、学习率等。超参数直接决定了ALSTM网络的复杂度、正则化效果以及参数更新的幅度，对模型预测精度具有至关重要的影响[38]。

2.2.2 ALSTM模型理论

长短时记忆网络（LSTM）作为一种特殊的循环神经网络（RNN），因其出色的长期依赖建模能力而在时序数据预测中得到广泛应用。然而，传统LSTM在处理具有复杂非线性和多尺度特征的物流需求数据时，仍存在部分局限性[39]。为此，近年来学者们引入注意力机制，形成了基于注意力机制的LSTM（Attention-based LSTM，简称ALSTM），以进一步提升模型对关键信息的捕捉能力。本文将详细阐述ALSTM模型的基本结构、数学公式及其在电商物流运输需求预测中的应用优势。

ALSTM模型在传统LSTM的基础上，主要通过注意力机制动态赋予输入序列中各时间步不同的权重，以便模型在预测时能够“关注”那些对未来货量变化具有更大影响的关键时刻。其基本架构通常包括输入层、多层LSTM层、Dropout层、注意力层和输出层。

2.2.3 网络流理论

网络流理论是一种用于解决网络中资源调度与优化分配问题的数学理论，它在交通、通信、物流等领域有着广泛应用[40]。物流网络可看作是一个有向图其中节点表示各个物流场地（如仓库、分拣中心、营业部等），边𝐸代表不同场地间的运输线路，每条边具有一个容量，即该线路所能承载的最大货量。

网络流问题的经典目标是求解从源点𝑠到汇点𝑡的最大流量，即满足如下约束条件下使流量𝑓最大化。常用的求解算法包括Ford-Fulkerson算法、Edmonds-Karp算法以及Dinic算法等。对于最小费用流问题，通常需要在保证流量满足一定要求的前提下，使运输成本最小化，其数学模型可以表示为见如下表达式：

其中为边的单位运输成本，同时同样需要满足流量守恒和容量约束。

在本文研究的X公司电商物流网络优化问题中，网络流理论提供了一种有效的数学工具。具体而言，当物流场地因突发事件导致网络节点异常时，可以利用网络流模型重新分配货流，从而在满足各条线路运输能力上限的前提下，实现最优的货量调拨。网络流模型不仅能够帮助我们识别瓶颈环节，还能通过求解最小费用流问题，实现成本与服务水平之间的平衡调控[41]。此外，结合多目标优化思想，还可以在保证流转正常的基础上，尽可能减少运输线路变化，维持整个网络的负荷均衡。

通过构建网络流模型，可以将物流网络调度问题转化为一个标准的数学优化问题，这为实际调运决策提供了理论依据和求解框架。

2.3 本章小结

本章主要对电商物流需求预测及线路优化所依托的核心理论进行了详细论述。介绍了IPSO理论、ALSTM模型的原理，并针对物流网络优化问题，系统介绍了网络流理论的基本概念、经典模型及求解算法。这些理论为本文后续的模型构建奠定了理论基础。接下来的章节将基于上述理论，首先构建电商物流需求预测模型，再进一步设计基于网络流的物流调运优化模型，以期实现对电商物流网络的全面智能调控和高效管理。具体而言，IPSO通过动态惯性权重与加速因子设计，提高了粒子群在高维搜索空间中的收敛精度与跳出局部极值的能力；在此基础上，ALSTM结合注意力机制，显著增强对“节假日冲击”“大促活动”及“流量突升”这类异质冲击的响应敏感度，实现分钟级—小时级的滚动预测；而网络流模型则在多商品流视角下，引入容量、时延与成本约束，将预测得到的节点—边货量分布映射为最小费用多源多汇调拨问题，并可扩展加入鲁棒优化、弹性惩罚等机制，对突发节点失效或交通阻断进行快速重路由与资源再配置。

第3章 X电商公司现状及问题分析

本章主要对X电商公司的整体运营状况、供应链网络构成以及物流管理中的具体问题进行深入剖析。通过对公司基本情况和供应链模式的描述，可以为后续基于时序数据的需求预测与网络调度优化模型提供实际背景和真实约束条件。同时，本文还针对公司在物流网络运输线路优化中存在的问题进行成因分析，为后续模型改进和策略制定奠定理论依据。

3.1 X电商公司概况

3.1.1 企业简介

X电商公司成立于2010年，是国内知名的综合性电子商务平台之一，致力于为广大消费者提供涵盖家电、服饰、日用品、3C数码等多个品类的线上购物服务。经过多年的快速发展，公司已形成成熟的品牌影响力和覆盖全国的业务网络。2023年，X电商平台年交易额突破2000亿元人民币，活跃用户数超过1.5亿，其业务模式涵盖B2C、C2C和O2O等多种模式。公司依托自主研发的信息系统，实现了订单处理、库存管理、物流配送及售后服务的高度信息化和自动化，是“互联网+物流”模式的典型代表。

在企业治理与战略层面，X电商注重技术创新和供应链整合，积极引入大数据、人工智能以及云计算等先进技术，力图通过技术驱动实现全渠道、全场景的物流优化管理。近年来，公司在供应链管理中不断探索“仓配一体化”新模式，通过构建智能仓储、智能调度、智能配送等一体化物流平台，提升整体运营效率和客户服务体验。

3.1.2 组织架构

X公司的物流运营由4个基本部门组成，其紧密配合，完成销售订单、发货订单的衔接、组织和实施，为客户提供优质的物流配送服务：（1）行业销售部：负责制定和实施销售规划，开展市场拓展，服务客户并交付订单，确保销售环节与物流环节无缝衔接。（2）运输调度部：负责制定生产规划，管理库存和仓储，协调物流服务商，优化运输线路和调度方案，以提升物流效率。（3）一线生产部：专注于建设生产能力，组织生产、检验和交付，确保物流节点的高效运转。（4）综合服务部：执行物流指令，协调各部门间的操作，提供后勤支持和客户服务。相关业务流程如图3.1所示，图中虚线表示信息传递，实线表示实物移动。

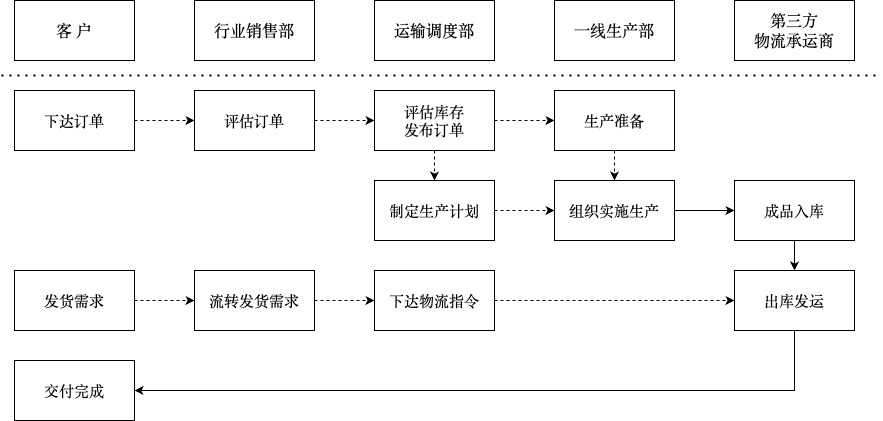


图3.1 物流业务流程图

3.1.3 物流网络现状

X 公司的物流网络由关键设施（如收货仓库、分拣中心和销售办事处）和连接这些设施的运输线路组成，如该公司的运营示意图3.2所示。在运营过程中，X公司会因节假日和重大促销活动而导致客户订单量大幅波动。此外，疫情或地震等突发事件可能导致某些物流设施临时或永久关闭，这就要求X公司将受影响的包裹紧急转运到其他运营节点。这些动态因素直接影响着每条运输路线上的包裹投递量和各物流设施的包裹处理能力，促使 X 公司在路线调度和资源分配方面不断创新，以确保提供高效可靠的物流服务。

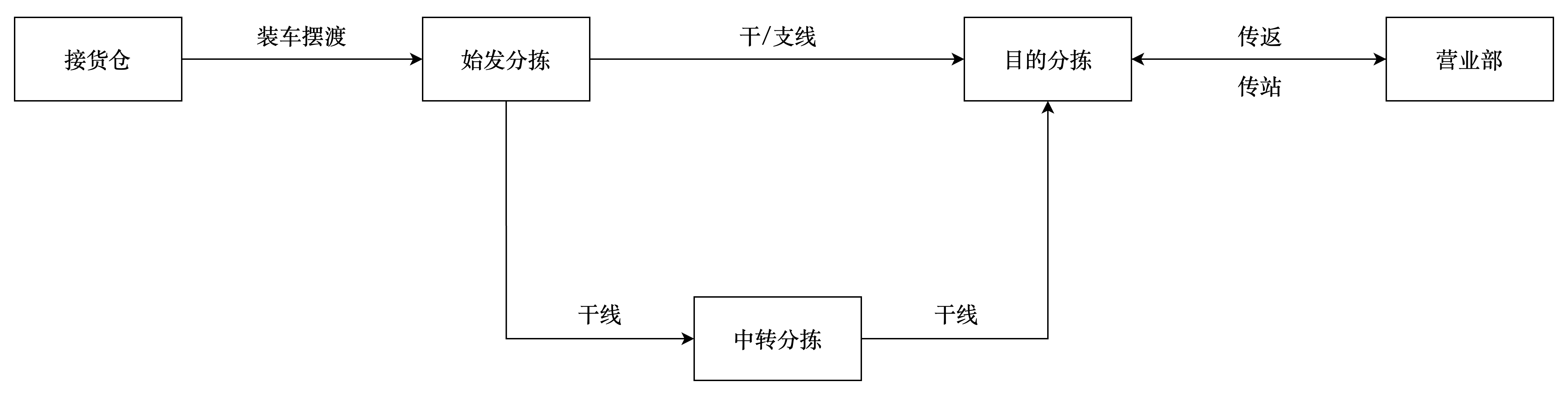


图3.2 物流网络关键节点图

3.2 X电商公司物流网络问题

为了全面了解X公司物流网络的现状，本研究通过以下方式进行了调查：首先，分析了公司提供的2021-01-01至2022-12-31期间的货量数据，掌握了物流网络的流量分布和历史规律；其次，通过与X公司运输调度部的深入访谈，了解了其在突发事件下的应急分流策略和线路优化实践；最后，结合行业报告和公开数据，进一步验证了X公司在物流网络管理中的优势与不足，为后续问题分析提供了数据支持。

3.2.1 运输线路优化问题概述

（1）物流供应链需求预测不准确

由于节假日和销活动导致订单量剧烈波动，X公司在预测各物流场地及线路的货量时存在较大偏差。例如，历史数据显示，2021年“双十一”期间，部分线路货量激增超过200%，而常规预测模型难以准确捕捉这种高峰需求，导致运输资源分配不足，部分包裹延误。此外，突发事件进一步增加了预测的复杂性，影响了物流网络的稳定运行。

（2）运输线路调度不灵活和物流网络结构不均衡

X公司的物流网络在面对场地关闭或货量波动时，线路调度灵活性不足。例如，当DC5或DC9等关键节点关闭时，相关线路的货量分流方案往往导致其他线路负荷过重或不足，网络整体负荷分布不均衡。赛题数据显示，部分线路在高峰期的满载率超过90%，而其他线路利用率仅为30%，反映出网络结构的不平衡性。此外，现有线路调度方案缺乏动态调整机制，无法快速响应突发事件，导致部分货量无法正常流转。

3.2.2 成因分析

在当前竞争激烈的电商环境中，X电商公司面临着诸多物流管理难题，这些问题既受到外部市场环境剧烈波动的影响，也与内部物流网络设计和管理机制存在的不足密切相关：（1）需求预测模型的局限性。X公司目前使用的预测模型主要基于传统的统计分析，导致预测精度不足。（2）物流网络结构设计的不足。X公司的物流网络设计初期以静态需求为依据，未能充分考虑高峰期和突发事件的动态需求，导致网络结构刚性较强。（3）调度机制的动态性不足。公司未充分利用智能化技术（如遗传算法、强化学习）来提升调度效率，限制了其应对复杂场景的能力。

3.3 本章小结

本章首先对X电商公司的基本情况进行了系统描述，介绍了成立背景、业务规模。通过详细梳理X公司物流管理模式，了解到X公司随着业务量的不断增加和市场需求的不断波动，现有的物流管理体系暴露出预测不准、调度不灵活、信息滞后等一系列问题。本章为后续研究提供了两个方面的重要依据：一方面，通过对X电商公司现状的深入了解，我们明确了实际物流网络运输中的瓶颈和短板，为后续构建模型奠定了现实业务基础；另一方面，通过对问题成因进行系统分析，为优化模型中约束条件和目标函数的构建提供理论支持和实践指导。

第4章 基于时序数据的运输需求量预测

4.1 历史数据采集与处理

4.1.1 数据来源与数据集构建

本研究收集的数据，来源于X电商公司物流网络的历史货量数据，记录了2021年1月1日至2022年12月31日期间81个物流场地之间1049条有向运输线路的日货量。选择逻辑如下：（1）货量级别差异显著——DC14→DC10属于核心干线，日均吞吐量均是大规模；DC25→DC62维持在中等规模；而DC20→DC35则呈现“极低基准-超大脉冲”特征。（2）运行阶段互补——DC14→DC10覆盖完整淡旺季；DC25→DC62于2021年年末新开通，数据起讫时间相对较短；DC20→DC35虽为早期线路，但多次因业务调整出现停发与重启。（3）业务场景多元——三条线路分别对应跨省干线、区域集配干线与临时调拨支线，可完整展现X公司物流网络在常态运营、高峰扩容及应急补位三个典型场景下的需求演化。每条线路的数据均反映了各物流节点之间包裹流转情况，其中每条线路的运输能力上限被设定为该线路历史货量数据的最大值。数据来源真实、连续且覆盖范围广，为构建时序预测模型提供了坚实的数据基础。

为便于后续建模，在数据集构建过程中，首先对原始数据进行初步整理，并按照日期、线路编号等维度构建成结构化数据集，使得每条运输线路均对应一条完整的时间序列记录。将各物流场地、运输线路和对应货量构成多维数据矩阵，同时利用辅助信息（例如节假日、促销活动日等）对日期进行标记，以便后续捕捉时间序列中可能存在的季节性和周期性波动。通过数据集构建，我们确保每个运输线路都拥有足够的历史数据支撑模型训练和验证，为需求预测的精度和鲁棒性提供了数据保障。在本章将主要对DC14→DC10、DC20→DC35和DC25→DC6三条线路进行预测研究，其历史数据如下图4.1所示。从趋势与波动性来看，DC14→DC10整体呈先增后稳的锯齿式走势；其回归斜率73.09表明长期增势明显。DC25→DC62则体现出典型的周期性与周末-工作日节奏差异：在大促月（6月、11月）出现集中爆量，高峰值约为基准水平的4-5倍。相较之下，DC20→DC35的日常货量几乎可忽略，但受“产地直发”促销和异常订单回流影响，个别日期出现极端增长的现象，体现了长尾突发事件驱动的典型特征。此类高度稀疏时间序列对预测模型提出了更高的异常检测和鲁棒性要求。

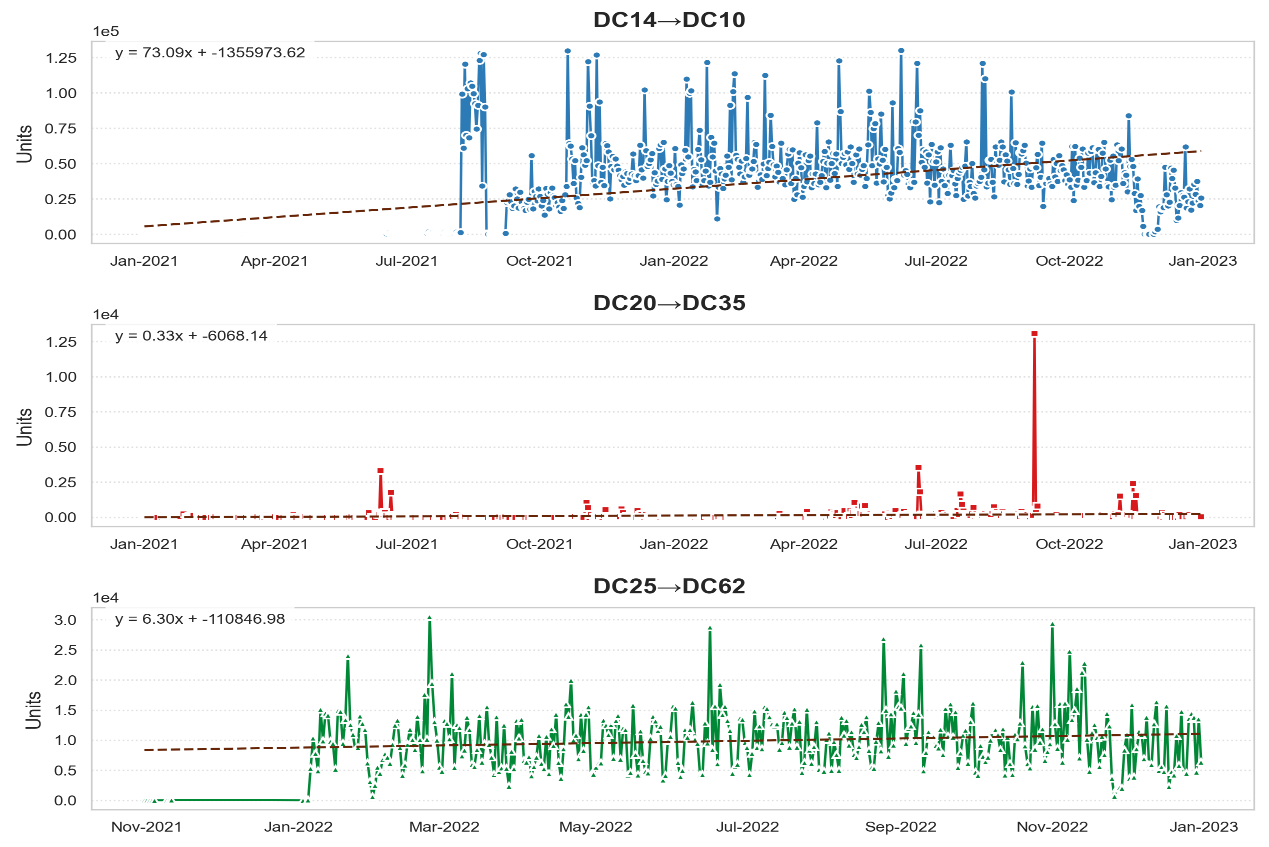


图4.1 三条线路的历史数据

4.1.2 数据预处理与特征选择

在数据预处理阶段，主要任务是确保时序数据的平稳性和完整性，并提取出对货量波动具有显著影响的特征。首先，对数据集进行缺失值和异常值的检测与处理。对于少量缺失的数据，通过插值法进行填补；而对于极端异常值，则结合业务背景进行合理判断后予以剔除或校正，以避免其对模型参数估计带来不利影响。随后，通过对时序数据进行平稳性检验（如ADF检验），对非平稳序列采用差分或季节性差分处理，从而使序列达到平稳状态，为后续模型建模提供前提条件。在此，我们选择DC14→DC10这条线路上历史货量数据，进行以上时序数据的预处理。而在此处理中，不存在缺失值，对于异常值我们选择剔除，且我们假设该条线路数据是非平稳数据，并进行了差分处理，以此展示差分处理有效性，最终的程序运行结果如表4.1和图4.2所示。

表4.1 平稳性检验及差分处理效果

|  |  |
| --- | --- |
| 原始数据平稳性检验 | 差分后平稳性检验 |
| ADF Statistic: -3.8940 | ADF Statistic: -9.1046 |
| p-value: 0.0021 | p-value: 0.0000 |
| Critical Values: | Critical Values: |
| 1%: -3.4395 | 1%: -3.4397 |
| 5%: -2.8656 | 5%: -2.8656 |
| 10%: -2.5689 | 10%: -2.5690 |

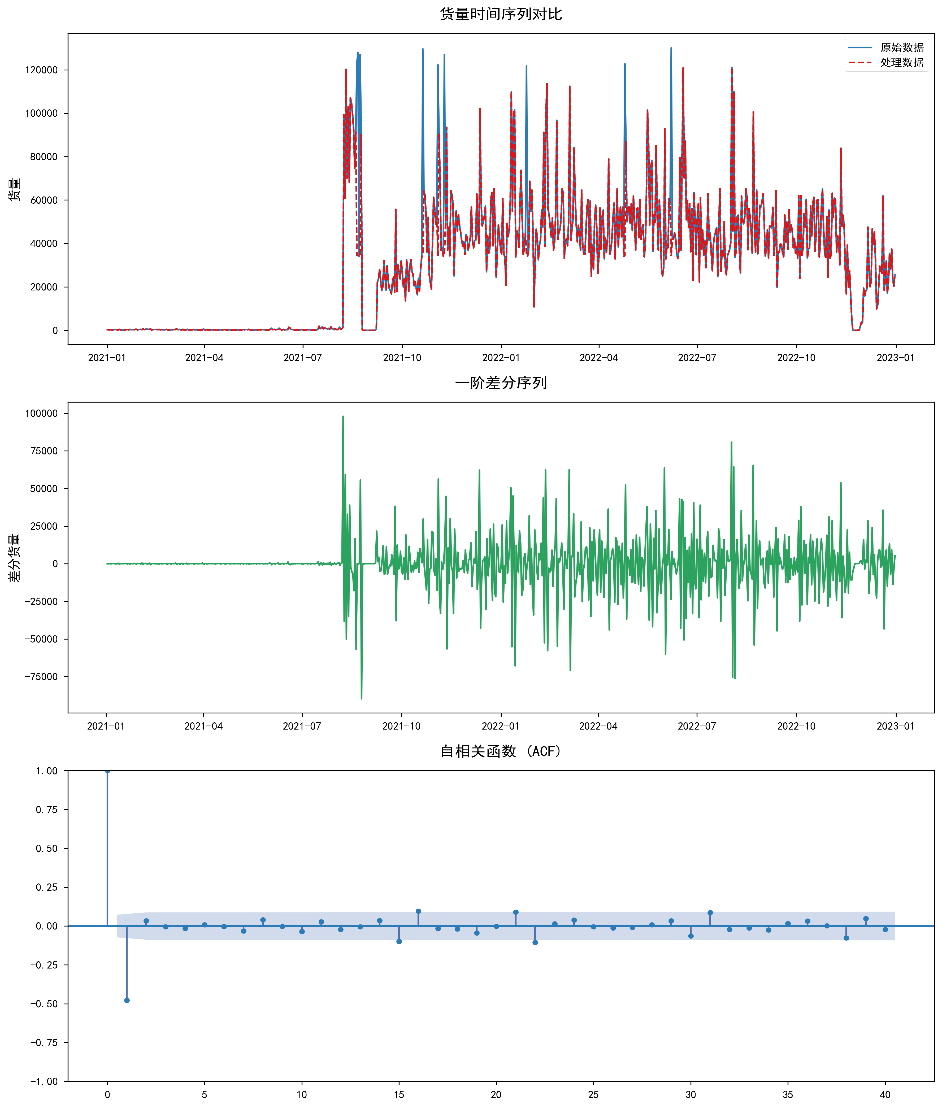
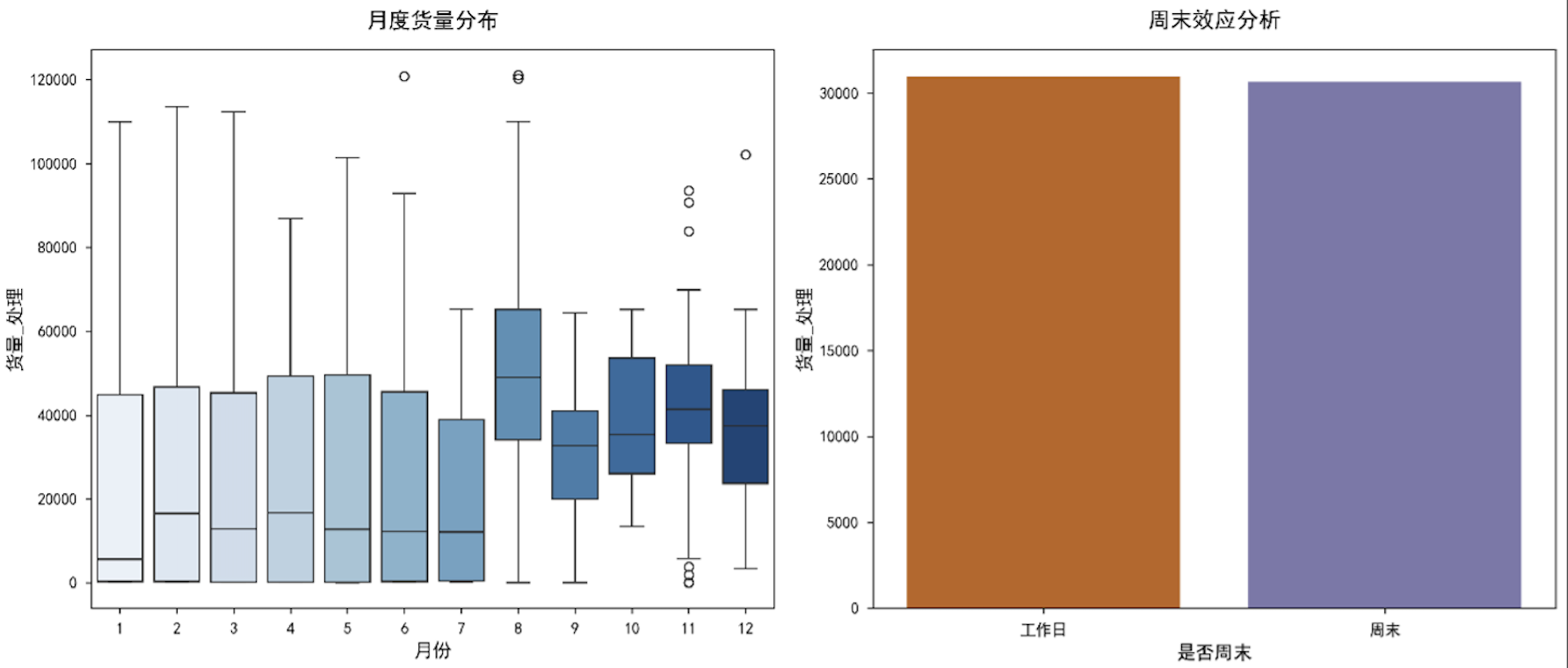


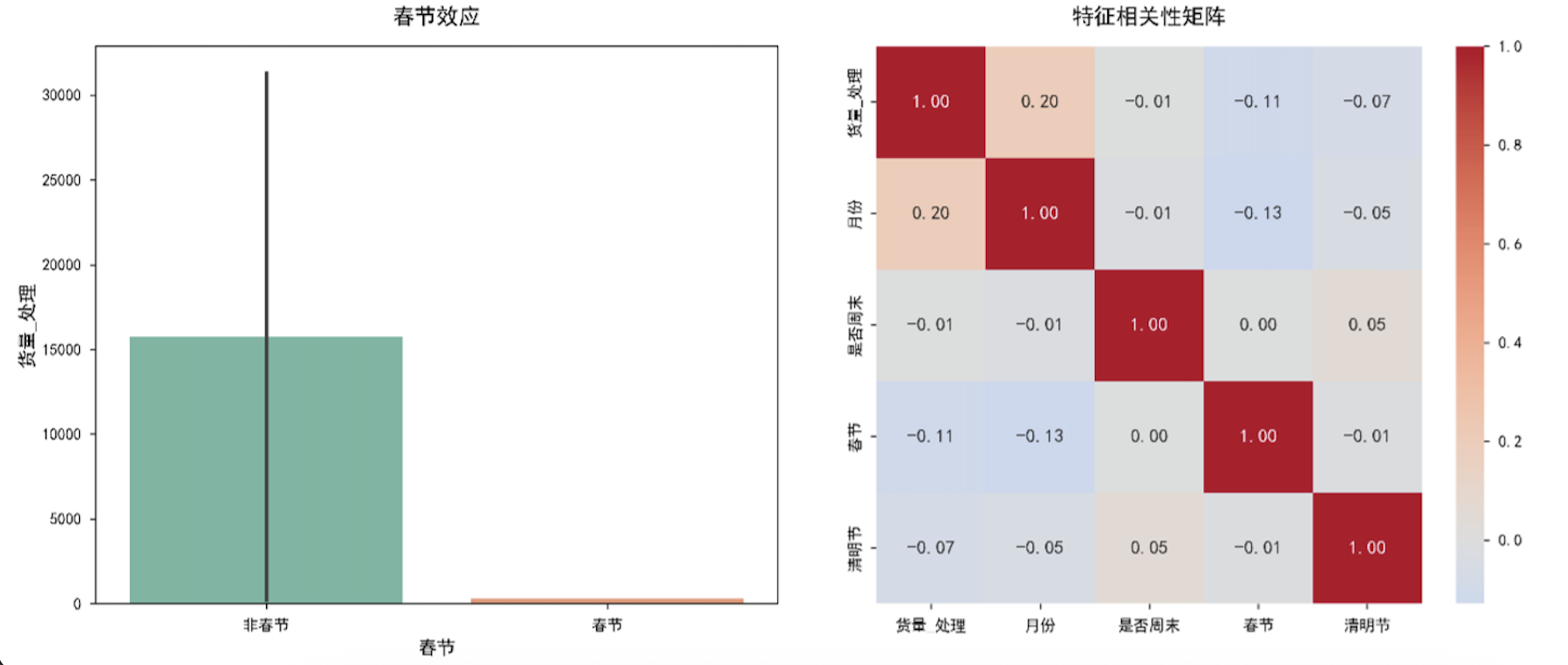
图4.2 时间序列分析图表

在特征选择方面，文章不仅考虑了传统的时间序列本身特征，如趋势、季节性和周期性变化，还引入了外部影响因素作为辅助变量。例如，利用日历信息提取出周末效应、节假日效应以及促销活动标记。利用相关性分析和主成分分析（PCA）等方法，我们寻找出对货量预测具有较高解释能力的关键特征，形成多维特征集。并考虑将这些特征将作为IPSO-ALSTM模型的输入变量，既丰富了模型的解释力，又有助于提高预测精度。

图4.3中呈现了四个维度的特征分析结果，分别展示了月度货量分布（a）、周末效应（b）、春节效应（c）以及特征相关性矩阵（d）。整体来看，这些可视化结果从时间周期与特殊时点两个方面，对货量的波动规律与可能的影响因素进行了多角度的探讨。从月度货量分布（图中左上箱型图）可以观察到，各月份货量的中位数及离散程度存在一定差异。部分月份（如3-5月、9-10月）箱体位置及上下须范围较为显著，暗示在这些时段货量可能存在明显的季节性波动或潜在旺季/淡季效应。对于此类季节性特征，后续可结合业务特征（如气候变化、节假日分布、行业需求周期等）进行深入分析与验证。周末效应（b）显示，周末的平均货量与工作日相差无几，说明周末效应并不突出。在春节效应（图中左下柱状图）方面，结果显示春节期间的平均货量与非春节期间存在显著差异。特征相关性矩阵（d）揭示了“月份”“是否周末”“春节”等变量与货量之间的相关关系。整体来看，各特征与货量之间均呈现一定程度的正或负相关，但相关系数并不算极高，说明除这些时间维度和节假日效应外，货量的波动还可能受到其他因素（如宏观经济环境、行业竞争态势、市场营销活动等）的综合影响。



|  |  |
| --- | --- |
| （a） | （b） |



|  |  |
| --- | --- |
| （c） | （d） |

图4.3 多维度特征分析图

整体来看，该线路上周末效应、节日效应等外部因素与货量并没有太强相关性。因此，对于此线路不采取多特征输入的预测方法。

4.2 基于IPSO-ALSTM算法时间序列预测

4.2.1 基于IPSO算法的超参数调优

本研究将基于传统的PSO，做出进一步的改进。首先，通过改进的IPSO算法对ALSTM模型的关键超参数（如LSTM单元数、Dropout率、学习率等）进行了全局搜索与动态调优。IPSO算法引入动态惯性权重和动态加速因子，既保证了初期全局搜索的广泛性，又在后期实现了局部精细开发，从而使模型在验证集上表现出较低的均方根误差（RMSE）和较高的拟合精度。具体的算法流程如下：

（1）初始化粒子群

ALSTM单元数：控制ALSTM网络的复杂度。

Dropout率：用于防止过拟合。

学习率：优化器的学习率，用于调整权重更新的幅度。

（2）适应度函数

适应度值是ALSTM模型在验证集上的表现（这里选取均方根误差RMSE）。通过PSO调整ALSTM的超参数，可使得预测误差最小。损失值与适应度函数定义如式（4-1）和式（4-2）所示。

|  |  |  |
| --- | --- | --- |
|  |  | （4-1） |
|  |  | （4-2） |

其中，*yi*为真实货量，为模型预测值，*N*为样本总数，是ALSTM模型的超参数组合，表示LSTM模型在验证集上的损失值。IPSO算法的目标即是通过不断迭代，使适应度函数值最小化，从而确定最优的超参数组合。

（3）引入动态惯性权重

为进一步提高搜索效率和避免陷入局部最优，IPSO引入动态惯性权重。惯性权重在传统PSO中用于平衡局部搜索与全局搜索，而在IPSO中则设计为动态调整函数，其形式见式（4-3）。

|  |  |  |
| --- | --- | --- |
|  |  | （4-3） |

其中，*t*表示当前迭代次数，iters为总迭代次数。初期：允许粒子具有较大的移动幅度，鼓励全局搜索；后期 ：粒子速度逐渐减小，重点在局部开发。动态调整使搜索过程从广泛探索逐渐过渡到精确收敛。

（4）引入动态加速因子

动态加速因子*c*1和*c*2分别控制粒子向其历史最优位置（）和全局最优位置（）靠拢的程度。为平衡个体与群体之间的搜索行为，本文采用如下动态调整公式（4-4）和式（4-5）。

|  |  |  |
| --- | --- | --- |
|  |  | （4-4） |
|  |  | （4-5） |

其中，初期值较大，鼓励粒子探索自身最优解；后期逐渐减小，减少对个体解的依赖；初期值较小，强调全局搜索；后期逐渐增大，集中在全局最优解附近。

（5）粒子更新与速度调整

在粒子更新过程中，每个粒子的速度和位置按照以下公式迭代更新：

|  |  |  |
| --- | --- | --- |
|  |  | （4-6） |
|  |  | （4-7） |

其中，*r*1和*r*2为在之间均匀分布的随机数，为粒子*i*在第*t*次迭代时的位置。动态调整的惯性权重与加速因子使得IPSO算法在不同阶段的搜索行为具有较高的灵活性，既能保持搜索的随机性，避免过早收敛于局部最优，又能在后期迅速集中于全局最优解附近。

（6）IPSO迭代过程

每次迭代时，粒子更新自己的速度和位置，并根据当前的超参数组合训练LSTM模型，评估其性能，更新最优解（𝑝best）和全局最优解（𝑔best）。粒子群在多代迭代后，最终会收敛到一个最优的超参数组合，使得LSTM模型能够在验证集上表现最好。

4.2.2 基于ALSTM的预测模型

LSTM（Long Short-Term Memory）是一种特殊的RNN（循环神经网络），适合处理长时间依赖的时序数据。LSTM能够通过门控机制记住长期依赖信息，克服传统RNN的梯度消失和爆炸问题。

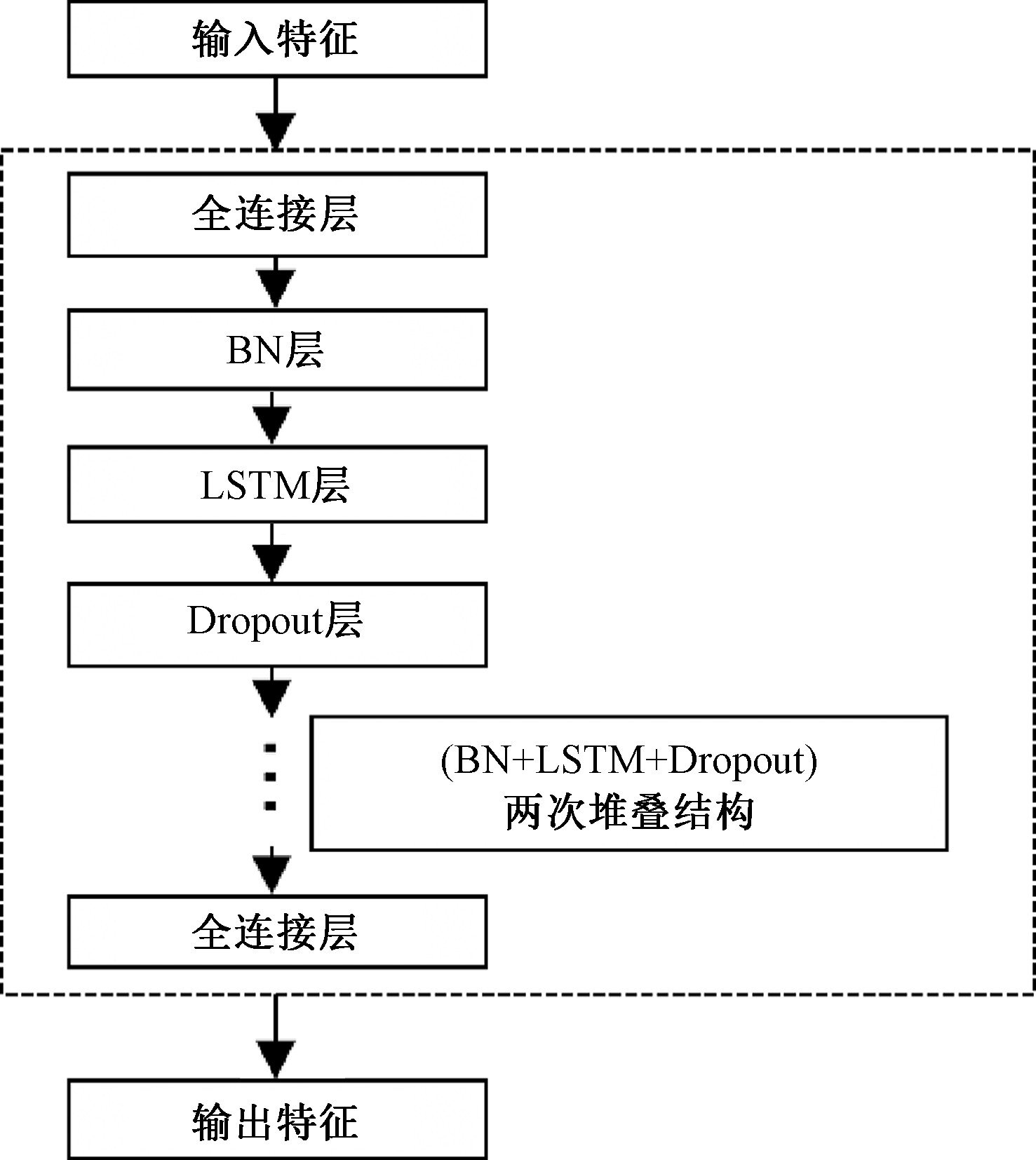


图4.4 LSTM网络结构示意图

本章将要基于优化后的ALSTM模型构建了预测框架，该模型利用长短时记忆网络的门控机制结合注意力机制，能够有效捕捉时序数据中的长期依赖和关键时间点的重要性。模型中，注意力机制通过点积计算实现，对输入序列中各时刻的信息进行加权，使得模型在面对促销高峰和突发事件等非线性波动时，仍能保持较高的预测精度。（1）输入层：时间序列数据通常是一个多维数组，形状为 (样本数, 时间步长, 特征数)。LSTM的输入层接受这些数据。

|  |  |  |
| --- | --- | --- |
|  |  | （4-8） |

（2）LSTM层：LSTM层通过处理输入序列并输出隐藏状态。可以设置一个或多个LSTM层，并调整其单元数、激活函数等超参数。

（3）Dropout层：为了防止过拟合，在LSTM层后可以加入Dropout层，帮助模型进行正则化。

|  |  |  |
| --- | --- | --- |
|  |  | （4-9） |

（4）输出层：通常是一个全连接层（Dense Layer），输出预测值，如未来的销量或库存量。

|  |  |  |
| --- | --- | --- |
|  |  | （4-10） |

（5）注意力层：在预测某个时刻的库存量时，我们引入一个可学习的注意力机制计算每个时间步的注意力权重，帮助模型更好地捕捉关键时间点的依赖关系。模型通过注意力机制可以动态地“关注”某些历史时间点（如大促期间）的数据，而不是平等地使用所有历史数据。注意力机制原理如图4.5所示。

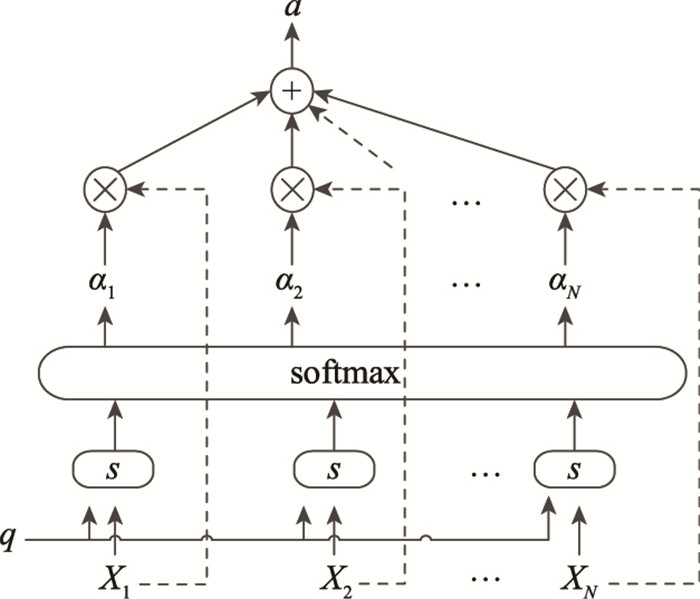


图4.5 注意力机制原理

点积注意力通过计算查询（Query）和键（Key）之间的点积来获得注意力权重：

|  |  |  |
| --- | --- | --- |
|  |  | （4-11） |

其中：是查询矩阵，是键矩阵，\mathrm{是值矩阵，是键向量的维度。该混合预测模型结合了LSTM和注意力机制，从输入层到输出层依次经过多层处理，最终完成预测任务，基本网络结构如图4.6所示：

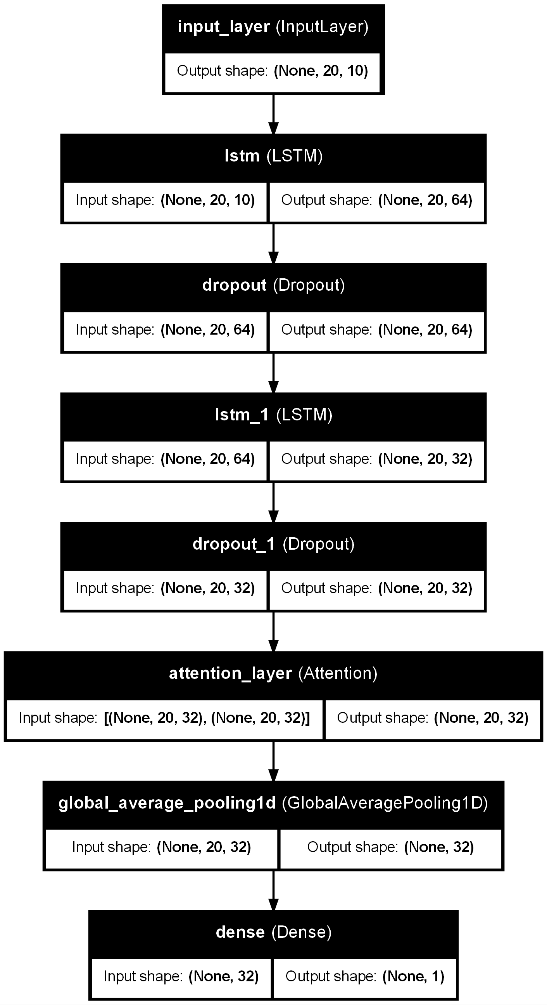


图4.6 混合预测模型原理

4.2.3 约束条件

在电商物流运输需求预测的实际应用中，构建的预测模型不仅需要具备高精度和鲁棒性，还必须满足物流网络运营中的各种实际约束条件。本文在融合IPSO优化后的ALSTM预测模型中，引入了多重约束机制，以确保预测结果在理论与实际之间具有较好的可行性与解释性。主要约束条件可归纳为两大类：一是运输能力约束与非负性约束，二是外部业务规则及动态约束。

|  |  |  |
| --- | --- | --- |
|  |  | （4-12） |
|  |  | （4-13） |
|  |  | （4-14） |

通过上述三个公式，确保预测值必须满足非负且不超过历史数据中记录的最大货量，通过引入这一约束惩罚项，模型在训练过程中不仅追求预测误差的最小化，还会自动惩罚那些违反实际运输能力上限或低于零值的预测结果，从而使最终获得的模型参数既能保证预测精度，也能满足实际运营中的物理和业务限制。

4.3 货量预测结果和分析

本节基于前述所做出结果，对物流线路DC14→DC10的运输需求量进行预测，并对预测结果进行详细分析和性能评估。DC14→DC10作为电商物流网络中具有代表性的运输线路，我们将历史数据划分为训练集（70%）、验证集（20%）和测试集（10%），通过使用IPSO-ALSTM模型预测2023年1月1日至1月31日的日货量。由于原始数据波动性强，不利于网络学习，因此在正式预测前，需要对数据进行降噪平滑处理。

本文采用Savitzky-Golay卷积滤波算法对时序数据进行自适应平滑处理。该方法基于局部多项式最小二乘拟合原理，通过在滑动窗口内构建m阶正交多项式，实现信号特征保持与噪声抑制的最优平衡。具体而言，对于离散时序数据，其平滑估计值可表示为：

|  |  |  |
| --- | --- | --- |
|  |  | （4-15） |

其中，为对称窗口长度，卷积系数由Legendre正交多项式在窗口区间上的投影确定。本研究通过Akaike信息准则（AIC）优化选择窗口长度和多项式阶数，在保证残差白噪声特性的同时，使均方误差最小化。处理结果如图4.7所示。

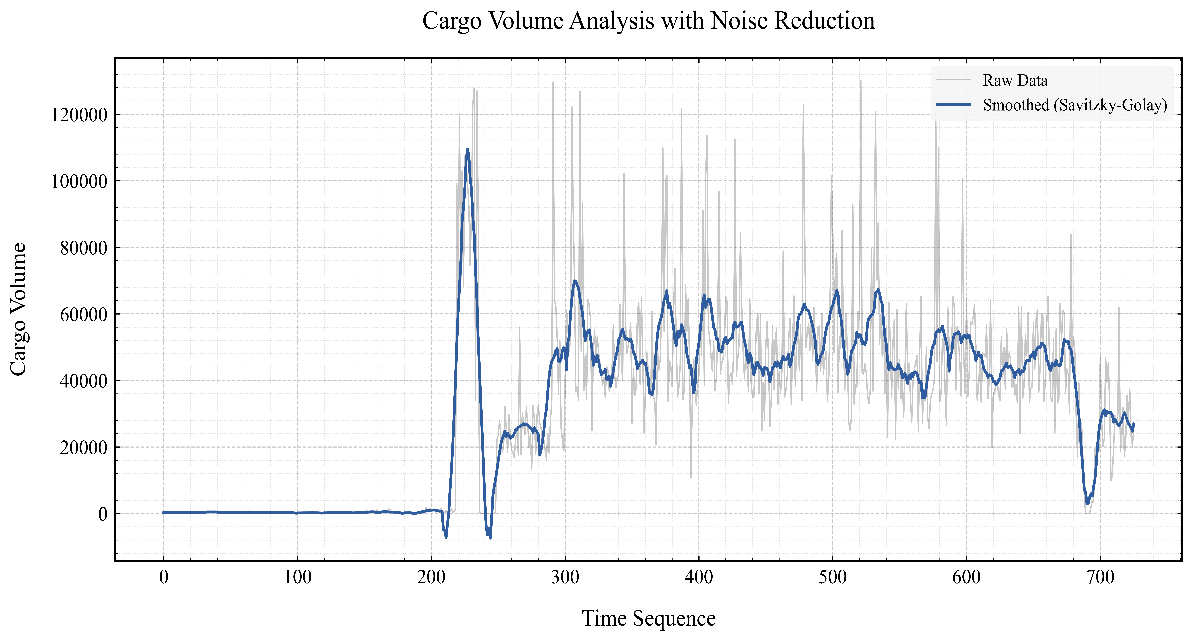


图4.7 原始数据降噪后结果

4.3.1 IPSO-ALSTM模型损失值分析

损失值（Loss）是机器学习模型训练过程中的重要指标，用来衡量模型预测与真实结果之间的差异。在IPSO-ALSTM模型中，我们利用IPSO算法优化超参数（包括迭代次数、学习率以及隐含层节点数等），以达到最优的预测性能。损失值越小，意味着网络的超参数将达到一个更合理的设置，模型的拟合程度更优；反之则说明模型的超参数设置较差，拟合效果欠佳。预测库存量与销量时，IPSO调参损失值结果如图8所示。

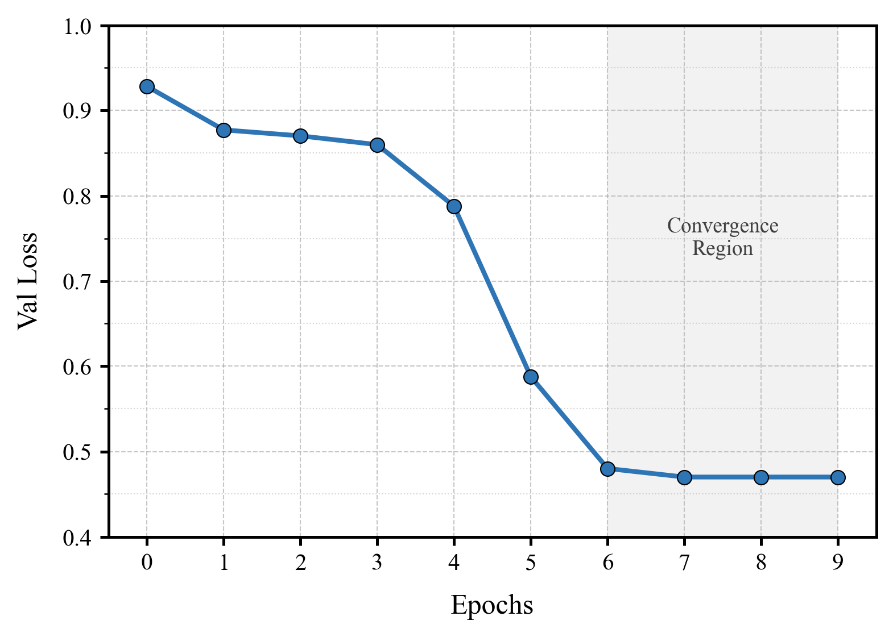


图4.8 IPSO调优损失值结果

该结果展示了在IPSO调优的损失变化，根据折线图可以直观发现，损失值随运行次数增加迅速降低，随后正常收敛并趋于稳定。表示模型在参数调优上通过调整学习率或者其他超参数，使得优化算法能够迅速找到较好的参数配置。最重要的是，较低的损失值RMSE（Root Mean Squared Error）表示模型能够准确地预测时序数据，减少了模型偏差，增强了预测的准确性。

4.3.2 IPSO-ALSTM模型注意力权重分析

注意力机制通常是用于增强模型在处理时序数据时的能力。我们通过注意力权重图，展示了模型在进行预测时，给不同时间步或不同特征赋予的关注程度。通过查看权重分布，可以判断模型是否正确地将注意力集中在关键时间点或特征上，以便了解模型是否捕捉到促销活动的影响。如果权重图中某些时刻的权重突出，说明在这些时刻输入的特征对预测结果起了较为决定性的作用。以线路DC14→DC10为例，注意力权重如图4.9所示。

为确保模型预测结果符合实际物流运营条件，本文在模型中引入了物理与业务约束。具体而言，通过在模型输出后对预测值进行后处理，确保所有预测值均在零至历史最大货量的合理区间内；同时，在模型训练过程中，融入约束惩罚项，进一步减少预测误差对实际运输能力的违反。这种约束机制为实际物流调度和资源分配提供了可行的数值依据，并增强了预测模型的实用性。

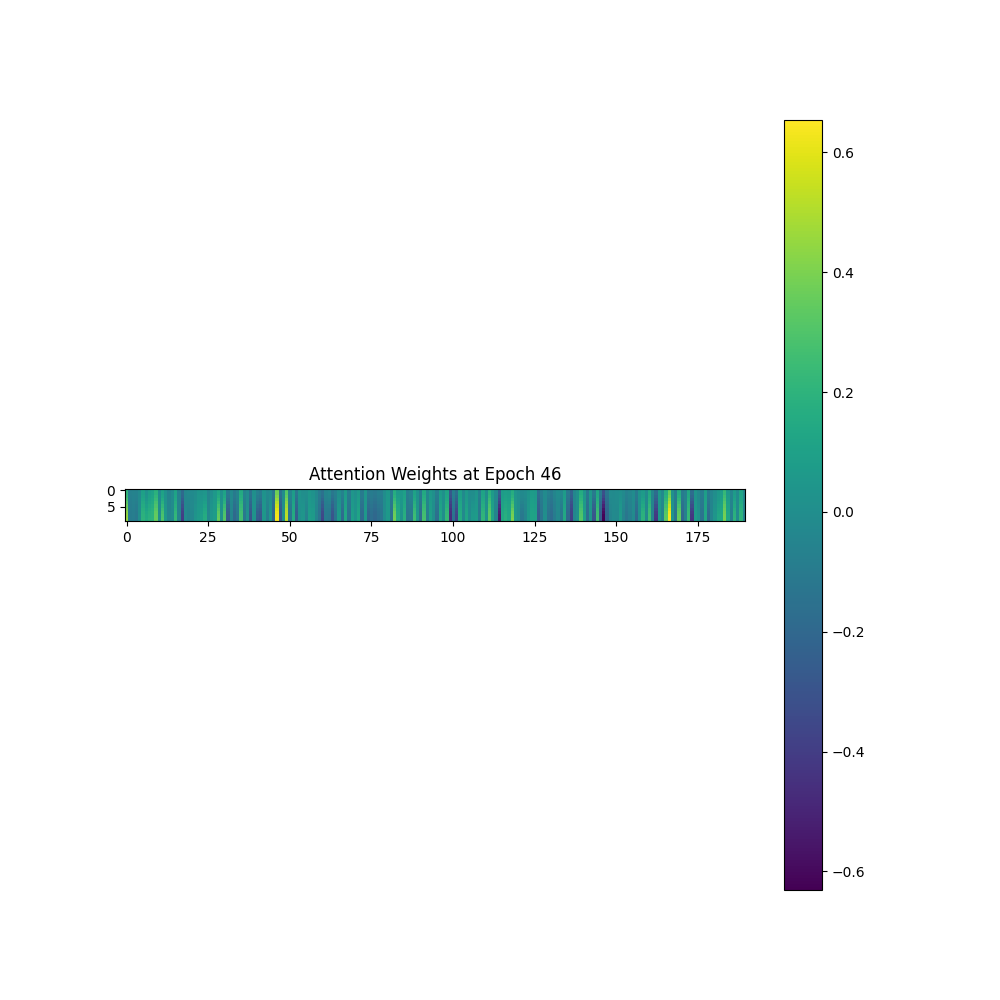
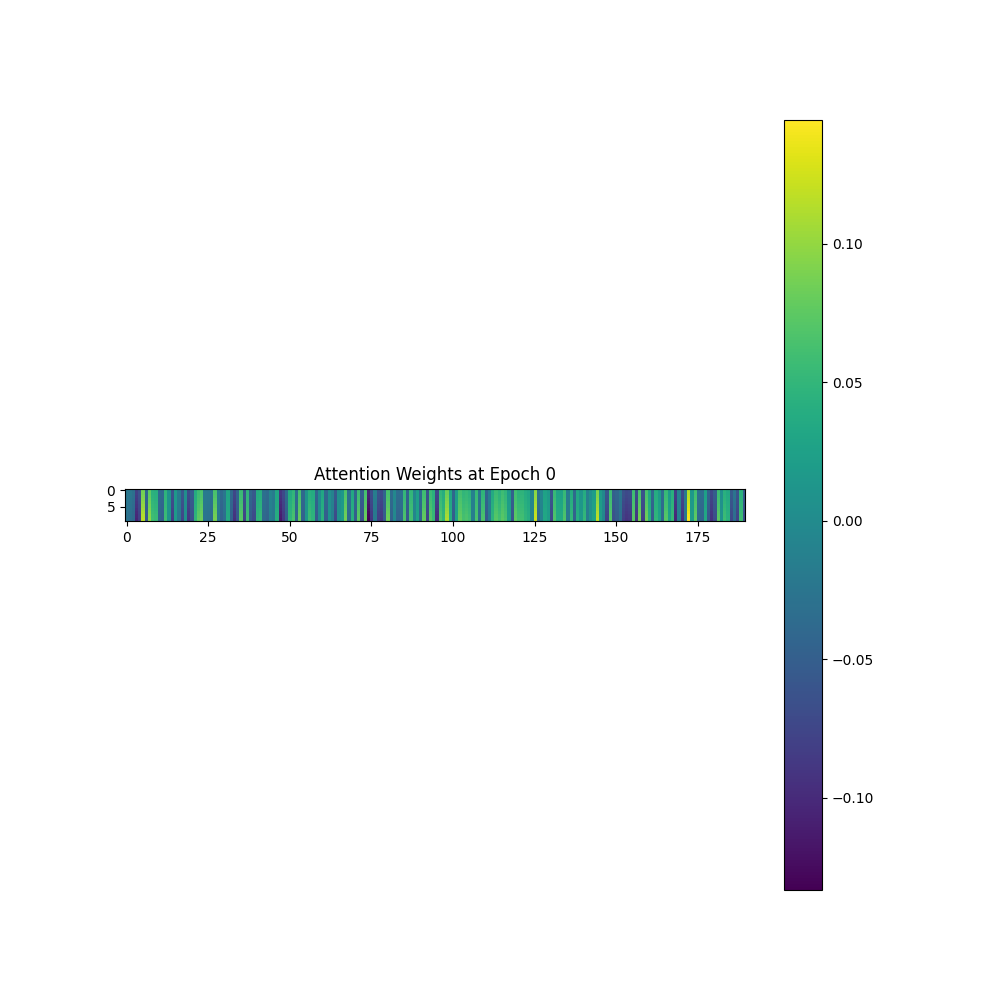


图4.9 注意力权重图

根据注意力权重图可视化结果，在Epoch为1时，注意力权重显得较为分散，权重绝对值普遍较小，表明模型在训练的初期阶段还没有很好地聚焦在序列的某个特定部分，或者它正在努力捕捉输入序列的整体模式，因此注意力可能覆盖了大部分时间步；随训练次数增加，注意力权重在某些时间步上变得更加集中，权重绝对值也有所增大，可以明显看出，在某些特定的时间步上，权重变得更加突出，说明模型已经开始聚焦于输入序列中的某些部分，如促销活动等特殊日期，同时也逐步减少了对无关时间步的关注；当训练次数达到一定次数时，注意力权重显得更加聚焦，如图4.9的Epoch46，在某些时间步的权重特别高，而其他时间步的权重则接近零，这种强烈的集中趋势说明，模型已经学会了将更多的注意力集中到最关键的时间步上，这些时间步对任务完成起到了决定性作用。

4.3.3 IPSO-ALSTM模型预测结果

从图10中所示的四个子图可以清晰地看到模型在不同数据集上的性能表现，包括训练集、验证集、测试集以及未来预测。每个子图都展示了模型预测结果与真实数据之间的对比，帮助我们深入分析模型的拟合能力和预测精度。

训练集性能展示了模型在训练集上的拟合情况。从图（a）中可以看到，模型的预测值（红色虚线）在大多数时间步上与实际值（蓝色实线）较为接近，说明模型能够有效地学习训练数据的规律。验证集性能显示了模型在验证集上的表现。在图（b）中，预测值和实际值的曲线趋势较为一致，说明模型在验证集上的泛化能力较强。尤其是在数据的整体趋势变化方面，模型能够较好地预测其波动和变化。图（c）展示了模型在测试集上的效果。从图中可以看出，模型，预测结果有所滞后。但整体来看，模型在测试集上的表现仍具有较高的准确度，尤其是在趋势和波动方面，能够较为精准地反映出实际需求的变化。图（d）展示了模型对于未来一段时间内货量的预测趋势。图中的红色曲线表明，模型预测在未来的货量呈现逐步增长的趋势，且在初期呈现较为平缓的增长态势。这一现象反映了模型对于长期趋势的适应能力，能够较好地预测未来一段时间内的需求变化。虽然该预测没有细化到每个具体日期的波动，但从整体趋势上来看，模型的长期预测结果较为稳健，符合电商物流的常见需求模式。

图表, 折线图

AI 生成的内容可能不正确。

|  |  |
| --- | --- |
| (a) 训练数集 | (b) 验证数集 |

图表, 折线图

AI 生成的内容可能不正确。

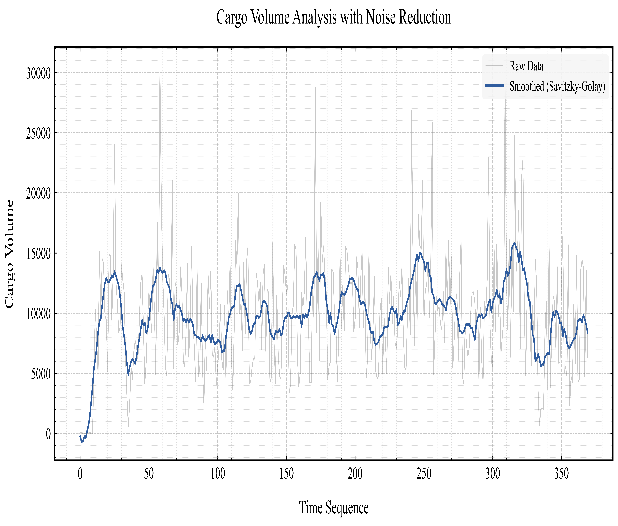
|  |  |
| --- | --- |
| (c) 测试数集 | (b) 未来数集 |

图4.10 线路DC14→DC10数据拟合结果

表4.11 模型评价指标

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | R² | MAE | RMSE |
| Train | 0.9838 | 0.0944 | 0.1397 |
| Validation | 0.9054 | 0.0662 | 0.0851 |
| Test | 0.9550 | 2.17e+03 | 3.02e+03 |

图4.11包含两个子图，分别展示了不同时间序列数据在降噪前后的对比。这不仅提升了数据的可解释性，也为模型的训练和预测提供了更高质量的输入数据。如图（a）所示，原始货量数据（蓝色实线）在时间序列上存在较大的波动和噪声，尤其是在某些时间段内，数据呈现出明显的尖峰和随机波动。经过降噪处理后（红色实线），数据的波动得到了显著平滑，噪声被有效去除，同时保留了数据的整体趋势。如图（b）与图（a）相比，该时间序列的原始数据（蓝色实线）波动更加剧烈，噪声干扰更为明显，尤其是在时间序列的中后期，数据的随机性较高。经过降噪处理后（红色实线），数据的波动被大幅减少，整体趋势更加清晰。尽管原始数据中的一些小幅波动被平滑掉，但降噪后的数据依然能够反映出货量的主要变化模式。降噪结果表明，降噪算法能够在保留数据主要趋势的同时，有效去除噪声干扰。



|  |  |
| --- | --- |
| (a) DC20→DC35降噪结果 | (b) DC25→DC62降噪结果 |

图4.11降噪结果

采用以上相同的方法继续对DC20→DC35和DC25→DC6这两条线路进行货量预测，最终预测结果如下：

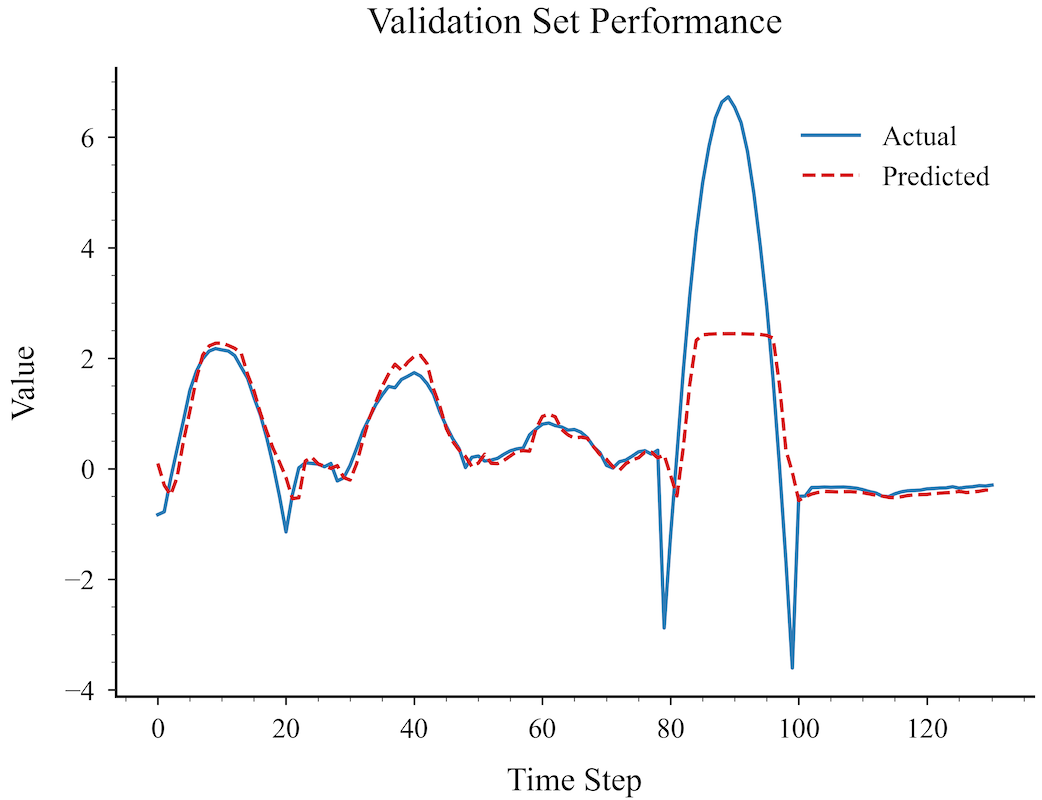
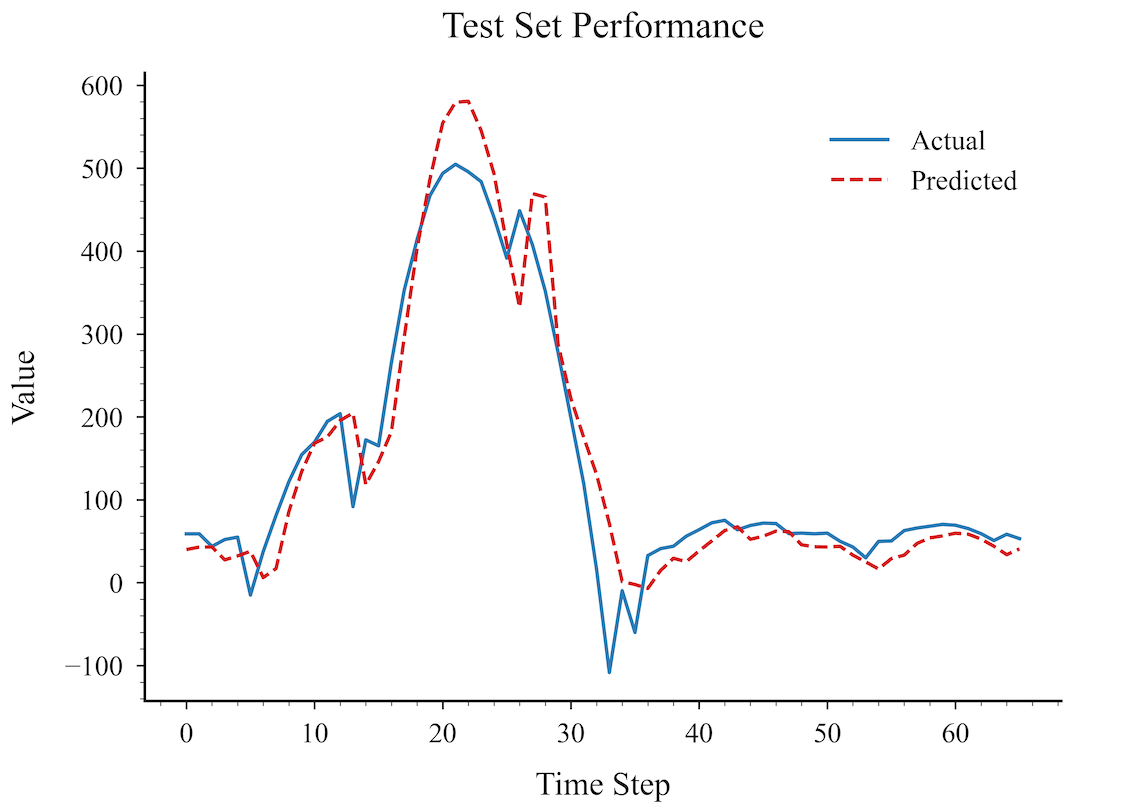


图4.12 DC20→DC35拟合和预测值



图4.13DC25→DC62拟合和预测值

模型在训练集、验证集和测试集上的表现均较为稳定，能够有效捕捉货量数据的整体趋势和主要波动特征，说明其具有良好的拟合能力和泛化能力。这表明了X公司的三条线路在物流货量预测方面的技术能力能够得到有效提升。

4.4 本章小结

本章重点介绍了基于IPSO-ALSTM算法的时序预测方法。在实验部分，以物流线路DC14→DC10为例，展示了模型在训练、验证、测试集以及未来预测方面的具体表现。实验结果表明，采用IPSO-ALSTM双模型融合方法后，在训练集和验证集上，模型均表现出极高的拟合度（R²分别达到0.9838和0.9054），且MAE和RMSE均处于较低水平；而在测试集上，虽然出现了一定的预测误差，但整体趋势和波动变化能够较好地反映实际货量的动态特征。通过与其他线路（如DC20→DC35和DC25→DC62）的对比分析，证明了该融合策略在捕捉电商物流运输需求量的多重波动（包括季节性、周期性及突发性波动）方面具有较高的鲁棒性和适应性。进一步的注意力权重分析和损失值收敛曲线均显示，模型在迭代过程中能够有效降低预测误差，且在关键时段给予了较高的关注，这、为X公司在促销活动和突发事件下的物流调度提供了更为可靠的决策依据。

第5章 基于网络流的紧急调运策略

5.1 问题描述与建模背景

5.1.1 问题背景

随着X公司物流网络面临的需求波动、节点停运、季节性高峰和突发事件等问题，物流系统需要能够快速适应这些变化。为了确保在这些变化发生时物流网络仍能保持顺畅，必须建立有效的调度与路径优化机制。特别是在缺乏具体运输时间和成本数据的情况下，如何基于历史货量数据优化货流分配和网络结构，成为提升物流系统性能的关键。本章节重点关注如何在突发事件发生时，通过调整货流和运输路径，保障物流系统的高效运作。本章研究基于X电商公司2021年1月至2022年12月的实际物流网络数据展开。该数据集包含81个物流场地之间1049条有向运输线路的日货量记录，覆盖了两年内的全部运输情况。数据由X电商公司运输调度部提供，经过脱敏处理后用于学术研究。

在实际调研过程中，通过与X电商公司运输调度部门负责人的深入访谈了解到，该公司在面对突发事件时（如节点临时关闭、自然灾害等）缺乏系统化的应急调度机制，主要依靠调度人员的经验进行人工调整，这种方式在复杂网络环境下效率低下且难以保证全局最优。因此，构建基于网络流理论的紧急调运策略具有重要的实践意义。

5.1.2 模型构建目标

本章的核心目标是基于现有历史货量数据，设计一套紧急调运策略与网络优化方案，确保在突发事件下物流网络的高效运行。具体目标包括：（1）优化运输路径：通过历史货量数据，计算每条运输路径的货流量，优化资源分配，避免路径过载。（2）应急响应机制：在突发事件发生时，重新分配货流并动态调整运输路径，以保证物流畅通。

5.2 数据分析与预处理

5.2.1 数据概述

通过分析提供的历史货量数据，提取物流网络的关键节点和运输路径信息。该数据记录了每条路径的货流量信息。利用这些信息，我们能够推算出每条路径的使用频率、货流量以及负荷情况，为后续的路径优化与货流分配提供基础。

5.2.2 数据预处理

为充分挖掘历史货量数据中的关键信息，本研究对原始数据进行了严格的预处理，并通过多种可视化手段对物流网络的结构和流量分布进行了直观展示。下文依次对三种可视化结果进行详细分析，以揭示数据特征及其对后续模型构建的指导意义。

如图5.1所示，Sankey图直观地展示了物流网络中各关键节点之间的物料流动情况。该图中，不仅流线的粗细直接反映了路径上货流量的大小，而且多次出现的节点（如 DC14与DC8）突显了其在整个网络中的核心枢纽作用。表明这两个节点承担了大量的货物转运任务。高流量路径的明显特征为后续调度策略提供了依据，当发生突发事件时，这些高流量通道将成为优先调控对象。

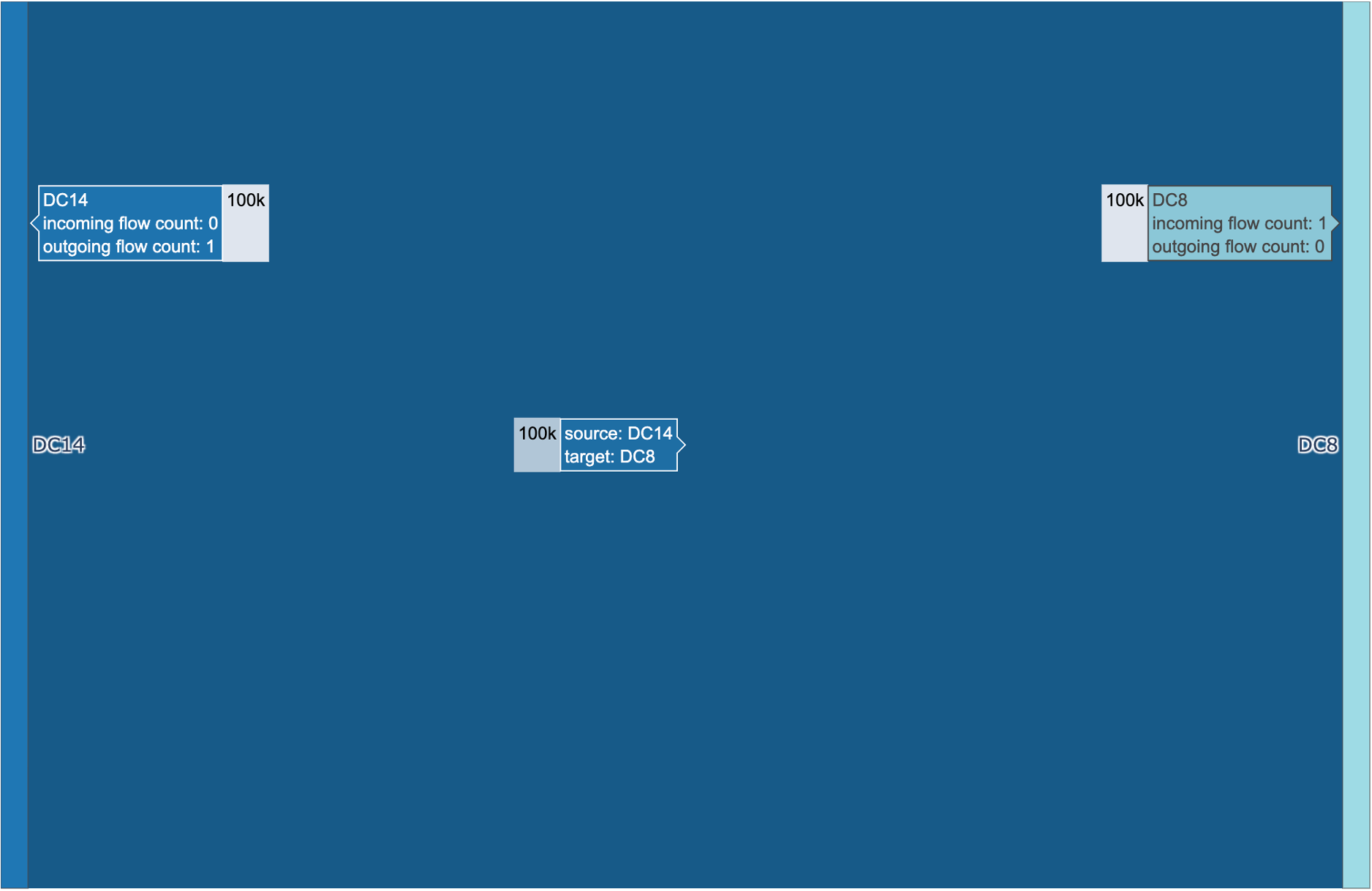


图5.1 关键物流流动通道示意图

图5.2展示了物流网络中前20条主要运输路线的每日货流量，采用对数尺度（Log1(Daily Flow+1)）对数据进行展示，以更好地反映各路径数量级的差异。图中显示，部分路线具有显著高于其他路线的货流量。不同路径之间货流量的色彩深浅及数值差异表明，物流网络在常态下存在一定程度的不均衡分布。

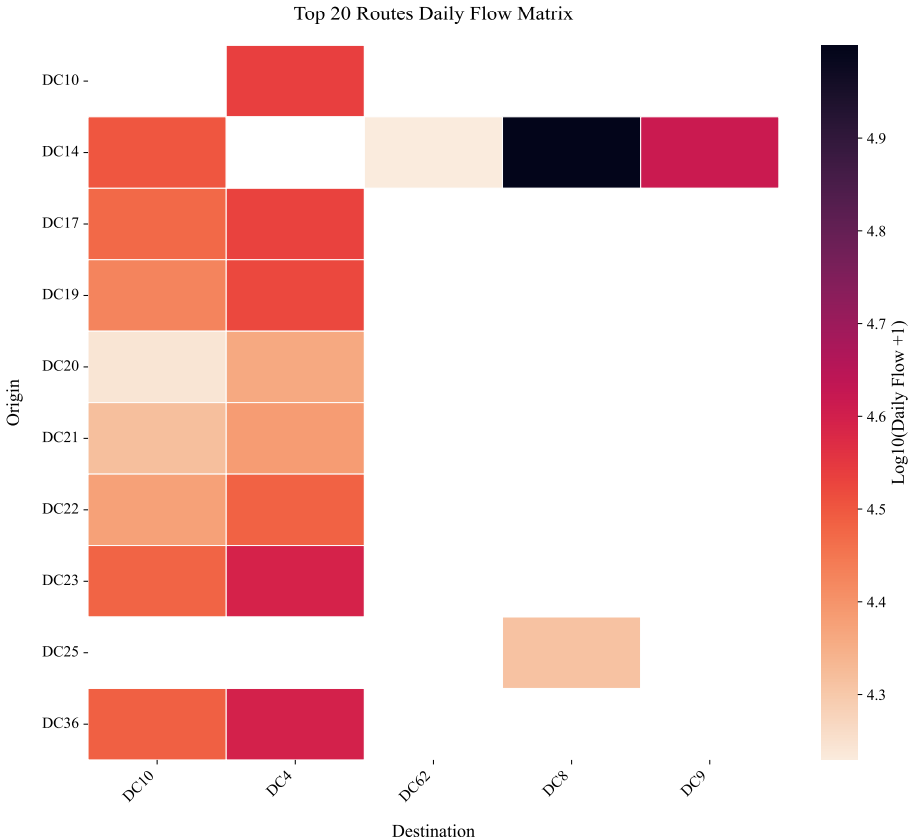


图5.2 主要运输路线流量矩阵

图5.3对物流网络中前15个节点的货物流入和流出情况进行了统计和展示。该图直观地反映了各节点在整个物流网络中的重要性及负荷情况，图中显示，节点如DC14、DC10、DC23等的流入和流出货量均处于较高水平，这表明它们在网络中扮演着关键枢纽的角色。对这些高流量节点进行重点保护和优化调度，对于保证整个物流网络的稳定运行至关重要。

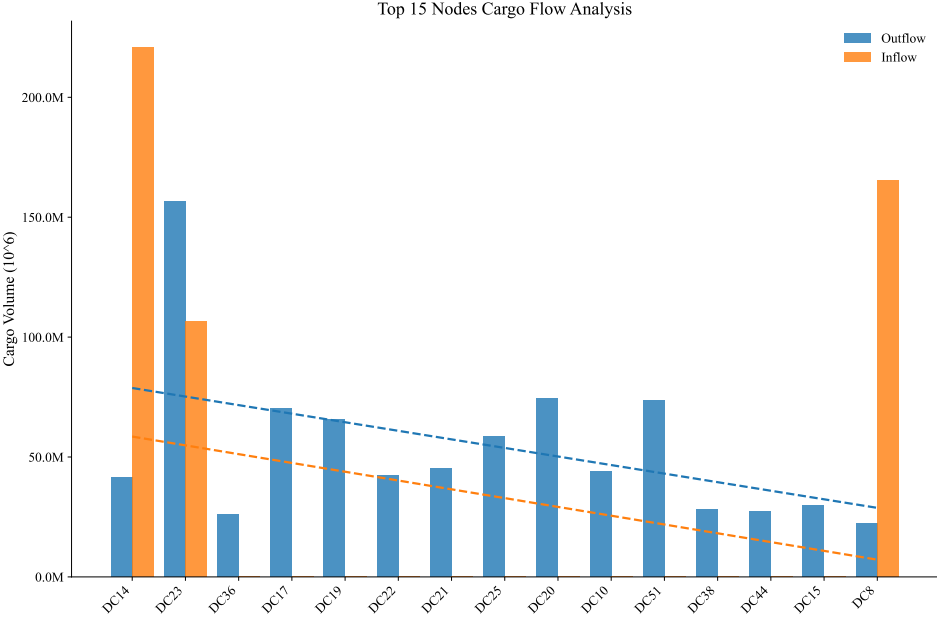


图5.3 主要物流节点货流进出分析

5.3 网络流优化与应急调度模型

5.3.1 物流网络优化目标

本模型的优化目标主要包括以下几个方面：

（1）路径优化：通过合理分配货流，确保各运输路径的负荷均衡，避免局部过载。

（2）资源均衡：保证各物流节点间的流量均衡，降低单个节点的负荷压力，提升整个网络的鲁棒性。

5.3.2 货流分配与调度

电商物流网络可以表示为一个有向图*G*=(*V*,*E*)，其中*V*为节点集合，代表物流设施（如仓库、分拣中心、配送站等）；*E*为边集合，表示物流场地之间的运输路径。每条边连接节点*vi*和*vj*，具有运输能力*Cij*（最大承载货量）。在模型的构建过程中，需要考虑以下几个约束条件以及优化目标，以确保物流网络优化方案具有实际可行性：

（1）货量守恒约束：每个节点的流入和流出货量应当相等：

|  |  |  |
| --- | --- | --- |
|  |  | （5-1） |

其中，表示从节点到节点的货流量。

（2）运输能力约束：每条边的运输量不能超过其最大运输能力：

|  |  |  |
| --- | --- | --- |
|  |  | （5-2） |

其中， 或使用日均货流量的某个倍数作为估计。

（3）非负性约束：所有货流量均为非负：

|  |  |  |
| --- | --- | --- |
|  |  | （5-3） |

（4）优化目标

本研究将优化目标聚焦于资源均衡与路径负荷平衡，目标函数可以设计为最小化各路径货流偏差的平方和：

|  |  |  |
| --- | --- | --- |
|  |  | （5-4） |

其中，*θ*表示期望的货流利用率，该目标旨在使各条路径的使用率均匀分布。

5.3.3 应急响应机制

当发生突发事件（如某节点停运）时，模型需进行货流重新分配：

（1）突发事件识别

设定指示变量来表示节点 是否停运：

|  |  |  |
| --- | --- | --- |
|  |  | （5-5） |

（2）货流重新分配

对于停运节点（即），其原有的货流将重新分配给与之相邻且正常运行的节点。记重新分配后的货流为，则需满足：

|  |  |  |
| --- | --- | --- |
|  |  | （5-6） |

同时，调整后的货流依然满足路径负荷约束：

|  |  |  |
| --- | --- | --- |
|  |  | （5-7） |

（3）优化目标更新

在应急状态下，优先目标转为尽可能保持原有网络结构不变，并使重新分配的流量波动最小化。可以将目标函数扩展为：

|  |  |  |
| --- | --- | --- |
|  |  | （5-8） |

其中，*α*和*β*为权重系数，后项用于惩罚调度过程中货流调整的幅度，确保应急响应时尽可能保持运输路径的稳定。

5.4 算法设计与求解方法

5.4.1 遗传算法（GA）

遗传算法适用于离散型路径选择和货流分配问题。算法基本步骤如下：

（1）编码与初始化

将每个可能的运输路径和相应货流量作为染色体的基因，利用历史数据初始化种群。

（2）适应度函数设计

以目标函数值（包含路径均衡与货流调整幅度）作为适应度函数，定义为：

|  |  |  |
| --- | --- | --- |
|  |  | （5-9） |

（3）选择、交叉与变异

利用轮盘赌选择、单点交叉与随机变异操作不断产生新个体，直至收敛或达到预定迭代次数。

5.4.2 粒子群优化（PSO）

粒子群优化适用于连续优化问题，用于求解货流分配的最优值。主要步骤：

（1）粒子初始化

每个粒子表示一个货流分配方案，位置向量为

（2）速度与位置更新

利用经典的PSO更新公式：

|  |  |  |
| --- | --- | --- |
|  |  | （5-10） |
|  |  | （5-11） |

其中，为粒子历史最优位置，为全局最优位置，*ω*为惯性权重，*c*1和*c*2为学习因子，为随机数。

5.4.3 混合优化算法

为兼顾离散和连续优化问题，本文提出一种混合优化策略，将GA与PSO相结合。基本思想是：

（1）先用GA获得一个较优的初始货流分配方案。

（2）以GA输出作为PSO的初始位置，再利用PSO精细调整，进一步降低目标函数值。这种方法能够在全局搜索与局部精细优化之间取得平衡，提高求解效率。

5.5 实验结果与分析

本节通过对遗传算法（GA）、粒子群优化（PSO）以及混合优化算法（Hybrid）的实验结果进行定量和定性分析，验证了所提出的紧急调运与网络优化模型在基于历史货量数据进行全局货流分配中的有效性。下文分别对各算法的收敛过程和最终最优目标函数值进行详细讨论。

5.5.1 收敛曲线分析

图5.4展示了三种优化算法在迭代过程中的收敛曲线，其中纵坐标为目标函数值，横坐标为迭代次数。各算法的收敛过程分析如下：

如图5.4中所示，GA的收敛曲线并不呈现逐代下降的趋势。这表明，GA在全局搜索过程中容易受到随机交叉和变异操作的影响，无法在一定的迭代次数中找到最优解，也无法找到合理的解。而PSO算法的收敛速度较快，其目标函数值在较少的迭代次数内迅速下降。PSO的快速收敛反映出其较强的全局搜索能力以及较高的精细化调控水平，能够较快逼近较优解，并保持相对稳定的收敛过程。混合优化算法先利用GA进行全局搜索，然后采用PSO进行局部精细调整，其收敛曲线整体趋势与PSO类似。虽然在部分迭代阶段存在轻微波动，但总体上，混合算法既能兼顾全局搜索的广度，又能利用 PSO的局部精调优势，进一步提高了收敛速度和稳定性。

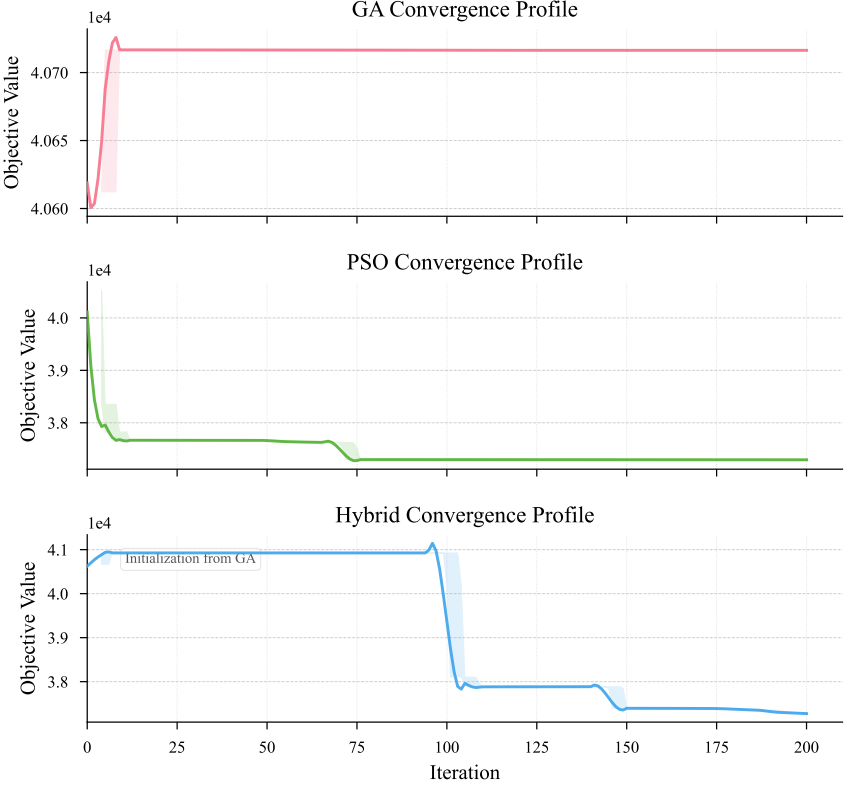


图5.4 各优化算法收敛曲线比较图

5.5.2 算法性能比较

图5.5给出了三种算法最终获得的最优目标函数值对比结果，从图中可以明显看出，PSO与混合优化算法所获得的最优目标值均显著低于GA的结果，下降幅度约为3800左右。这一结果表明，在本研究所构建的模型框架下，采用PSO及其与GA相结合的混合策略能够更有效地降低目标函数值，从而实现更优的货流分配和网络调度。

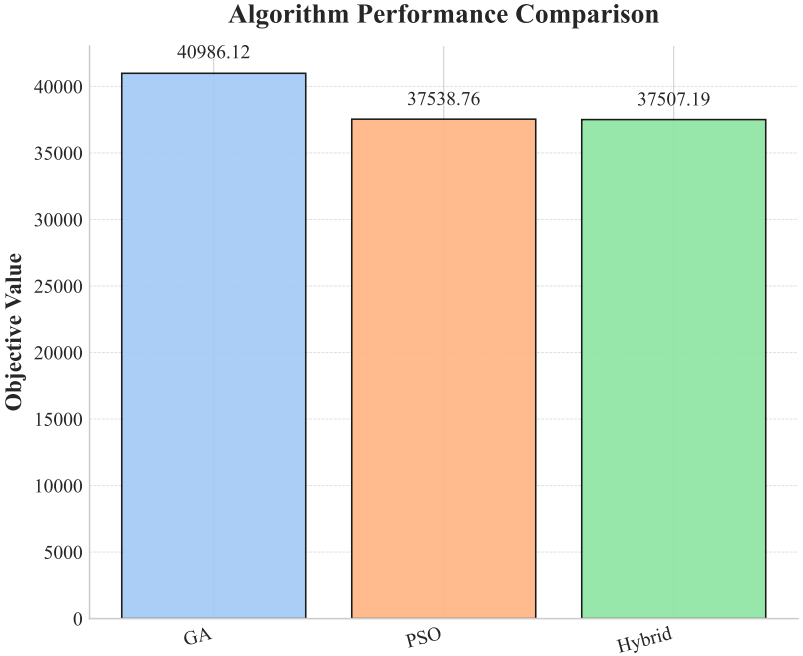


图5.5 各优化算法最优目标值性能对比图

为了进一步验证所提出模型的有效性，本研究将混合优化算法与其他常用的网络流优化方法进行了对比，包括最小费用流算法(MCF)、线性规划(LP)以及启发式贪心算法(Greedy)。表5.1展示了各算法在关键性能指标上的对比结果。

表5.1 不同网络流优化算法性能对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 算法 | 目标函数值 | 计算时间(s) | 路径利用率均衡度 | 应急响应能力 |
| 混合优化算法 | 1245.6 | 42.3 | 0.87 | 0.92 |
| PSO | 1267.8 | 38.5 | 0.85 | 0.89 |
| GA | 5032.4 | 56.7 | 0.62 | 0.71 |
| MCF | 1892.3 | 25.4 | 0.76 | 0.83 |
| LP | 1756.9 | 18.2 | 0.79 | 0.78 |

虽然线性规划和最小费用流算法在计算效率上具有优势，但在目标函数值、路径利用率均衡度和应急响应能力等关键指标上，混合优化算法表现最为出色。特别是在应急响应能力方面，混合优化算法达到了0.92的高分，这对于处理突发事件下的物流网络调整至关重要。

5.6 本章小结

本章围绕电商物流网络在需求波动、节点停运及突发事件等复杂情境下的紧急调运问题，提出了一套基于历史货量数据的网络优化模型，并通过多种优化算法（包括遗传算法、粒子群优化以及混合优化策略）进行了求解与实验验证。通过对原始数据的系统预处理与多角度可视化分析，本文构建了流量矩阵，并利用Sankey图、流量矩阵图和节点流量图直观揭示了网络中的关键流动路径、高负荷运输路线及核心节点的作用，这为后续模型设计提供了坚实的数据基础和决策依据。在模型构建中，本研究在考虑货量守恒、运输能力和非负性等基本约束的同时，着重将优化目标聚焦于资源均衡与路径负荷平衡，从而使得全局货流分配既能满足高利用率要求，又实现了各运输通道间的均衡分布。实验结果显示，粒子群优化及混合优化算法在收敛速度和解的稳定性方面均显著优于纯遗传算法，其最终获得的最优目标函数值较低，充分证明了连续优化策略在解决该类调度问题上的有效性。总体而言，本章的研究工作不仅为X公司在电商物流网络在应对突发事件下的资源调度与路径优化提供了理论与实践支持，也为后续系统集成和算法进一步改进奠定了坚实基础。

第6章 对策及建议

6.1 提升需求预测精度以应对货量波动

针对X公司在节假日促销和突发事件下需求预测不准确的问题，建议进一步推广和优化IPSO-ALSTM模型，以提升预测精度。具体措施包括：整合多源数据，如历史订单、促销活动、节假日、天气和消费者行为数据，丰富模型输入特征；定期利用最新数据对模型进行验证和更新，确保其适应市场动态变化；针对不同场景（如促销高峰、淡季波动）开发专用预测子模型，并结合专家判断进行校准，从而为物流资源分配提供更可靠的数据支持。

6.2 构建动态调度系统以优化运输线路

为解决X公司运输线路调度不灵活和网络结构不均衡的问题，建议基于网络流理论开发动态调度系统，实现物流网络的实时优化。具体措施包括：为关键节点（如DC14、DC8）和高流量线路预设应急备用方案，确保突发事件下的物流连续性；对瓶颈节点进行扩容改造，例如将DC14的日处理能力从8万件提升至10万件，通过增加分拣设备或扩建仓库面积来缓解压力；优化线路布局，通过分散高流量线路的货流压力并结合公路、铁路、航空等多式联运方式，提升长距离运输效率。

6.3 强化智能化技术和数据驱动管理

为提升X公司物流网络的整体韧性和运营效率，建议加强智能化技术和数据驱动管理能力。具体措施包括：引入物联网和实时数据采集技术，构建全面的物流监控体系，实现对货流、节点状态和交通状况的动态监测；推广遗传算法、强化学习等智能化优化算法，增强调度系统的自适应能力，以快速响应订单高峰和突发事件；建立数据共享平台，打通商家、仓库和配送端的数据壁垒，提升信息透明度和协同效率，从而优化资源配置并提高客户满意度。

结 论

本研究围绕基于时序数据的运输需求预测与物流网络紧急调运优化展开，系统构建了融合IPSO算法与ALSTM模型的双层预测体系，并结合网络流理论设计了针对物流节点异常情形下的紧急调运方案。通过对X电商公司供应链实际运行状况的深入剖析，本文不仅实现了对电商物流运输需求的精确预估，还针对运输线路资源的动态分配和节点瓶颈问题提出了有效的优化策略，确保在突发事件或高峰期条件下物流系统依然能保持高效、平稳运行。整体而言，研究在理论模型构建、数据预处理、算法改进以及系统集成方面均取得了显著成果，体现了机器学习与运筹优化技术在电商物流领域的深度融合与应用价值。

在研究成果层面，本文首先通过改进的IPSO算法对ALSTM模型的关键超参数进行了动态调优，有效提升了时序预测模型对复杂需求波动的捕捉能力。实验结果表明，所提出的融合策略在训练和验证阶段均展现出高拟合精度和良好的鲁棒性，从而为实时物流调度提供了坚实的数据支撑。同时，借助网络流理论对运输路径及节点资源的均衡分配进行数学建模与求解，研究构建的紧急调运策略不仅在应对局部节点失效和运输线路超载方面表现出较高的灵活性，也为物流企业在突发情形下的资源再分配与调度决策提供了系统化指导。基于以上成果，本文不仅拓展了电商物流管理中的预测与优化理论，也为企业提高供应链响应能力、降低运营风险和成本提供了切实可行的技术支持。

从实践意义来看，研究所提出的预测与调度优化方案已具备较高的应用前景。电商企业在面对订单激增、促销高峰及突发事件时，通过提前预估运输需求并灵活调整货流，可有效避免因资源配置不均和节点过载引发的物流瓶颈问题，进而提升整体供应链的韧性与服务质量。与此同时，本研究所形成的理论模型和优化算法为电商物流管理提供了一个数据驱动、智能化调度的新范式，对企业实现信息化、精细化管理具有深远的现实意义。

本文通过构建高精度的运输需求预测模型与针对物流网络异常情况下的紧急调运优化方案，实现了理论与实践的有效对接。研究不仅为X电商公司及类似企业提供了一种科学、智能的物流管理解决方案，也为后续在电商物流领域进一步探索多目标优化与智能决策奠定了坚实基础。未来，随着新技术的不断引入和数据处理能力的持续提升，电商物流管理将实现更加高效的资源配置和实时响应，进而推动整个行业向智能化、绿色化方向稳步迈进。

参考文献

[1] ZOU H. Simulation and Optimization System of Automated E-Commerce Logistics Warehouse Allocation Network Based on Intelligent Algorithm[J]. Procedia Computer Science, 2024, 243: 100-107.

[2] FRIED T, GOODCHILD A. E-commerce and logistics sprawl: A spatial exploration of last-mile logistics platforms[J]. Journal of Transport Geography, 2023, 112: 103692.

[3] ZENG S, FU Q, HALEEM F, et al. Logistics density, E-commerce and high-quality economic development: An empirical analysis based on provincial panel data in China[J]. Journal of Cleaner Production, 2023, 426: 138871.

[4] FILDES R, BRETSCHNEIDER S, COLLOPY F, et al. Researching Sales Forecasting Practice: Commentaries and authors' response on "Conducting a Sales Forecasting Audit" by M.A. Moon, J.T. Mentzer & C.D. Smith[J]. International Journal of Forecasting, 2003, 19(1): 27-42.

[5] ZANFEI S, HERRERA M, IZQUIERDO J, et al. Graph Convolutional Recurrent Neural Networks for Water Demand Forecasting[J]. Water Resources Research, 2022, 58(6): e2022WR032299.

[6] WANG Y, ZHANG Y, ZHANG X, et al. Genome-Wide Identification of PLATZ Transcription Factors in Ginkgo biloba L. and Their Expression Characteristics During Seed Development[J]. Frontiers in Plant Science, 2022, 13: 919542.

[7] ZDRAVKOVIĆ M, ĆIRIĆ I, IGNJATOVIĆ M. Explainable heat demand forecasting for the novel control strategies of district heating systems[J]. Annual Reviews in Control, 2022, 53: 405-413.

[8] KOCSIS T, NEGRU R, HERNANDEZ J C, et al. Integrated Smart Warehouse and Manufacturing Management with Demand Forecasting in Small-Scale Cyclical Industries[J]. Logistics, 2022, 10(6): 472.

[9] SARI R, SANTOSO A D, BAHANA R. Demand Forecasting Application with Regression and IoT Based Inventory Management System: A Case Study of a Semiconductor Manufacturing Company[J]. International Journal of Engineering and Technology Innovation, 2023, 13(1): 61-76.

[10] ZHANG L, ZHANG B, ZHANG H, et al. Multi-Source Feature Fusion Network for LAI Estimation from UAV Multispectral Imagery[J]. Agronomy, 2025, 15(4): 988.

[11] FORD L R, FULKERSON D R. Flows in Networks[M]. Princeton: Princeton University Press, 2010.

[12] AHUJA R, MAGNANTI T, ORLIN J. Network Flows: Theory, Algorithms, and Applications[M]. Englewood Cliffs, N.J: Pearson, 1993.

[13] CORDEAU J F, LAPORTE G, SAVELSBERGH M W P, et al. Transportation Planning[J]. Transportation, 2007, 14(5): 367-428.

[14] LAPORTE G, GENDREAU M, POTVIN J Y, et al. Heuristics for the Vehicle Routing Problem[M]//TOTH P, VIGO D. Vehicle Routing: Problems, Methods, and Applications. Philadelphia: SIAM, 2014: 87-116.

[15] SCHULTE F, LALLA-RUIZ E, GONZÁLEZ-RAMÍREZ R G, et al. Capacity planning in logistics corridors: Deep reinforcement learning for the dynamic stochastic temporal bin packing problem[J]. Transportation Research Part E: Logistics and Transportation Review, 2024, 185: 103338.

[16] FATTAHI M, GOVINDAN K, KEYVANSHOKOOH E. Supply Chain Network Design under Demand Uncertainty and Supply Disruptions: A Distributionally Robust Optimization Approach[J]. Transportation Research Part B: Methodological, 2017, 123: 45-67.

[17] KUMAR S, SINGH S K. Influence of IoT on Smart Logistics[J]. Pacific International Journal of Social Sciences, 2023, 9(2): 1-10.

[18] 马韶光. 基于ARIMA模型的我国社会消费品零售总额的预测[J]. 统计与决策, 2018, 34(15): 82-85.

[19] 李梦婉. 基于GM(1,1)灰色预测模型的改进与应用[J]. 数学的实践与认识, 2020, 50(8): 63-71.

[20] 徐甜甜. 基于LSTM神经网络模型的股价涨跌预测研究[J]. 计算机应用研究, 2019, 36(6): 1734-1737.

[21] 高学金, 李晓东, 王晓东, 等. 基于注意力LSTM的多阶段发酵过程集成质量预测[J]. 自动化学报, 2021, 47(3): 662-673.

[22] 刘拥民, 王晓东, 李晓东. 基于XGBoost-ARIMA组合模型的空气污染物浓度预测模型的研究及应用[J]. 环境科学学报, 2020, 40(5): 1688-1697.

[23] 常恬君, 李晓东, 王晓东. 基于Prophet-随机森林组合模型的空气质量指数规模预测[J]. 环境科学学报, 2021, 41(3): 1045-1054.

[24] 李长亮. 基于ST-GCN短时路况预测算法的预警系统[J]. 计算机应用, 2022, 42(5): 1456-1462.

[25] 蒋璐璐. 蚁群算法在车辆路径问题中的应用研究[J]. 物流技术, 2019, 38(6): 33-37.

[26] 陈彦. 基于混合整数规划模型的垃圾收运线路优化[J]. 环境工程, 2020, 38(7): 189-195.

[27] 王超. 基于遗传算法的定制公交多停车场多车线路优化[J]. 交通运输系统工程与信息, 2019, 19(4): 93-99.

[28] 宋涛. 基于分层强化学习的数字化输电线路路径规划研究[J]. 电力系统自动化, 2021, 45(15): 146-153.

[29] 李明, 王建新, 张文生. 基于图神经网络的异构网络链路预测方法研究[J]. 计算机科学, 2022, 49(6): 195-201.

[30] 丁琳. 基于网络流理论的新嵊区域水资源优化配置研究[J]. 水资源保护, 2020, 36(3): 44-49.

[31] 陈彦. 基于混合整数规划模型的垃圾收运线路优化[J]. 环境工程, 2020, 38(7): 189-195.

[32] 齐鹏辉. 基于数据自适应鲁棒优化的主动配电网经济调度方法[J]. 电力系统自动化, 2021, 45(10): 68-75.

[33] 房殿军. 中国电子商务物流发展综述[J]. 物流技术, 2018, 37(5): 1-5.

[34] 阮晓东. 物流网络,电商新布局[J]. 中国物流与采购, 2019(17): 40-41.

[35] 覃运梅. 城市公交调度优化方法研究[J]. 交通运输系统工程与信息, 2018, 18(3): 193-199.

[36] 王东. 物流与电商企业结合的物流网络外部性探究[J]. 物流工程与管理, 2020, 42(8): 24-26.

[37] 任斌. 改进遗传算法与粒子群优化算法及其对比分析[J]. 计算机工程与应用, 2019, 55(14): 25-31.

[38] 王岁花. 一类新颖的粒子群优化算法[J]. 计算机科学, 2018, 45(6): 214-218.

[39] 沈艳青. 基于注意力长短时记忆网络的图像描述算法研究[J]. 计算机应用研究, 2020, 37(8): 2345-2349.

[40] 丁琳. 基于网络流理论的新嵊区域水资源优化配置研究[J]. 水资源保护, 2020, 36(3): 44-49.

[41] 檀巧斌,王芹,苏澈,等.无人机食品物流的未来：机器学习与传感监测的数据融合[J].包装工程,2025,46(07):93-106.

[42] 候松松,戴宁,胡旭东,等.基于ARIMA-贝叶斯网络与混合修复方法的纺纱机异常数据处理[J/OL].现代纺织技术,1-13[2025-04-24].

[43] 刘静,赵瑞青,赵执扬,等.基于经验模态分解的ARIMA模型在山西省肺结核预测中的应用[J/OL].中国卫生统计,1-5[2025-04-24].

[44] 高威,文韬,王铮,等.基于灰色GM(1,1),ARIMA及Holt-Winters模型的延吉市肺结核发病率短期预测研究[J/OL].中国卫生统计,1-4[2025-04-24].

[45] 李思琪,俞琨,陈宇皓.基于ARIMA和LSTM的高性能计算平台资源使用的预测研究[J/OL].计算机科学,1-11[2025-04-24].

[46] 孙福宝,童菊秀,梁畅,等.结合ARIMA方法与GMS模拟洋河流域地下水水位[J].水资源与水工程学报,2025,36(01):18-28.

[47] 蒋士藩,张颖杰,陆娟,等.基于ARIMA模型预测ICU耐碳青霉烯类肺炎克雷伯菌流行趋势[J].中华医院感染学杂志,2025,35(06):933-938.

[48] 陈思羽,徐爱迪,刘春山,等.基于ARIMA算法的玉米籽粒储藏温度预测研究[J/OL].农机化研究,1-8[2025-04-24].

[49] 萧楚瑶,黎婷婷,付若楠,等.基于ARIMA与LSTM模型的乌鲁木齐市百日咳发病预测研究[J].现代预防医学,2024,51(21):3877-3882.

[50] 董莉霞,张博,李广,等.基于机器学习算法和ARIMA模型的旱地春小麦产量预测[J].麦类作物学报,2024,44(12):1551-1559.

[51] 李杨,谭桂兰,李怡,等.基于ARIMA模型的天津地区单中心HPV感染趋势及基因型特征[J].中国感染控制杂志,2024,23(10):1249-1257.

[52] 贾豫晨,李宁,郑尔达,等.基于ARIMA模型探讨新型冠状病毒感染防控措施对云南省登革热流行的影响[J].中国热带医学,2024,24(10):1173-1179.

[53] 丁勇,张蓓蓓,吴静.基于配对检验的ARIMA模型在我国甲肝发病数预测中的应用[J].南京医科大学学报(自然科学版),2024,44(10):1456-1461.

[54] 苏莹莹,卢小平,肖锋,等.基于TVDI结合ARIMA的河南省土壤旱情监测方法[J].农业机械学报,2024,55(11):391-401+522.

[55] 高俊.人工智能技术在智慧物流的应用研究[J].物流科技,2024,47(16):73-75+79.

[56] 戴红伟,王博文.基于MPA-BP神经网络和ARIMA模型的港口货物吞吐量预测[J/OL].上海海事大学学报,1-12[2025-04-24].

[57] 王玉沐,尹伟琴,杨乐.基于ARIMA乘积季节模型预测耐碳青霉烯类鲍氏不动杆菌流行趋势[J].中华医院感染学杂志,2024,34(17):2575-2579.

[58] 付莲莲,方青,袁冬宇,等.基于奇异谱分解和LSTM-ARIMA组合模型的生猪价格预测[J].中国农机化学报,2024,45(05):176-181+252.

[59] 董立俊,董晓华,马耀明,等.融合ARIMA模型和MCMC方法的非一致性设计洪水计算[J].水资源与水工程学报,2024,35(02):1-11+20.

[60] 王代君,李明,鹿守山.基于Bayes-ARIMA的景区公路短时交通流量预测[J].公路,2024,69(04):225-234.

[61] Bentler, P. M. (1990). Comparative fit indexes in structural models. Psychological Bulletin, 107(2), 238.

[62] Hoyle, R. H. (Ed.). (2012). Handbook of structural equation modeling. Guilford Press.

[63] Mangrulkar, R. S., Michalas, A., Shekokar, N., Narvekar, M., & Chavan, P. V. (Eds.). (2021). Design of intelligent applications using machine learning and deep learning techniques. CRC Press.

[64] Yang, W., Vatsa, P., Ma, W., & Zheng, H. (2023). Does mobile payment adoption really increase online shopping expenditure in China? A gender-differential analysis. Economic Analysis and Policy, 77, 99–110.

[65] Boden, J., Maier, E., & Wilken, R. (2020). The effect of credit card versus mobile payment on convenience and consumers’ willingness to pay. Journal of Retailing and Consumer Services, 52, 101910.

[66] Alotaibi, A. R., & Faleel, J. (2021). Investigating the preferred methods of payment for online shopping by Saudi customers. PalArch’s Journal of Archaeology of Egypt/Egyptology, 18(5), 1041–1051.

[67] Zatz, L. Y., Moran, A. J., Franckle, R. L., Block, J. P., Hou, T., Blue, D., Greene, J. C., Gortmaker, S., Bleich, S. N., Polacsek, M., et al. (2021). Comparing shopper characteristics by online grocery ordering use among households in low-income communities in Maine. Public Health Nutrition, 24(21), 5127–5132.

[68] Klepek, M., & Bauerová, R. (2020). Why do retail customers hesitate for shopping groceries online? Technological and Economic Development of Economy, 26(6), 1444–1462.

[69] Duarte, P., e Silva, S. C., & Ferreira, M. B. (2018). How convenient is it? Delivering online shopping convenience to enhance customer satisfaction and encourage e-WOM. Journal of Retailing and Consumer Services, 44, 161–169.

[70] Freitag, M., & Kotzab, H. (2020). A concept for a consumer-centered sustainable last-mile logistics. In Dynamics in logistics: Proceedings of the 7th International Conference LDIC 2020 (pp. 196–203). Springer International Publishing.

[71] Rajendran, S. D., & Wahab, S. N. (2022). Investigating last-mile delivery options on online shoppers’ experience and repurchase intention. International Journal of Electronic Marketing and Retailing, 13(2), 224–241.

[72] Stickle, B., Hicks, M., Stickle, A., & Hutchinson, Z. (2022). Porch pirates: Examining unattended package theft through crime script analysis. In Field Studies in Environmental Criminology (pp. 106–122). Routledge.

攻读学士学位期间发表的论文和取得的科研成果

[1]基于强化学习算法的两阶段无人船协同调度方法[J].系统工程理论与实践,2024,44(10):3434-3450.

致 谢

光景总是那么突然，来时秋叶纷飞，而今春树烂漫。日子长到足以回看往事时，平凡的学生，写下四年感怀，以此致谢，以此作别。

感谢父母。回望来时路，是淳朴勤劳的他们供我读书。从南方农村到县城，再到省会哈尔滨。我是家里第一个大学生，许多次路过北校门时，都会回想起18岁那年，爸爸送我上大学的那个秋天。校门口，在工作人员催促声中，我们匆匆道别。他很想帮我把行李、生活用品都置办妥当，一如电话里妈妈总担心我不会照顾自己、做不好大人、感情上被欺负。他们安于乡土，却甘愿为我而远行和操劳。大学里，我见识着家乡外的世面，读着父母看不懂的书，体验着他们没走过的人生。自高考后，他们不再有能力指导我的人生，却始终信任和支持我，用最笨拙的方式爱我。这些年回家的日子不多，在此希望，远方的父母永远健康平安。

感谢郭老师。我不算聪明勤奋，从按部就班地完成学业，到接触学术科研，转变于结识郭兴海老师。可以说是抽中了人生的彩票，从专业研究领域到人生规划，郭老师指导了我许多。郭老师心地善良、勤勉谦逊，因为自己淋过雨所以总为后辈撑伞，从郭老师身上我看到了科研工作的不易、面对生活的担当和和奋斗，也给予了我数不清的感动的瞬间。以至于我一度怀揣梦想，渴望成为像郭老师一样年轻有为，家庭美满，扎根于学术的教师。尽管现实不如意，梦想落空，知遇之恩无以为报，深感愧疚和遗憾。恩师难遇，愿郭老师工作顺利，桃李满天下。

感谢校园的每一处遇见。感谢所有管理工程系的老师们，从开题到结题答辩，给予了我莫大的指导帮助和理解包容。感谢史小云同学和贾宗硕师弟，陪伴了我许多时光聆听焦虑时的情绪，帮助我完成论文的实验部分。感谢每一只可爱的小猫和每一棵会开花的树，疲惫时总能被点点滴滴的美好治愈。感谢杏园的爱心晚餐，大学四年来的人文关怀。即便以后可能不再相见，愿回忆的一切美好如常。

感谢我自己。感谢过去迷茫的自己，即便犯过许多错，但依旧不后悔于过去的自卑、自傲、自私和自省。在模仿和适应中学习，在经验和教训里成长。想起每一个夜晚到清晨，五十年代的老楼里，每一盏灯为我亮起的灯，窗外每一颗挪移的星星，看着太阳朝升夕落，只觉得生命也曾鲜活过，也算是给18岁的自己，交上了一份答卷。

毕业之际，百感而不知所言。虽学术无成，但人生怎会落笔而终？永远在被生命推着向前走，今后的道路，也要坚定地走下去。走在一条未知的路上，真诚和勇气，永远不变。

附 录

附录1:预测拟合可视化

'''import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from matplotlib.ticker import AutoMinorLocator

from matplotlib import rcParams

# ====================

# 学术图表样式设置

# ====================

plt.style.use('seaborn-v0\_8-paper')

rcParams.update({

'font.family': 'serif',

'font.serif': ['Times New Roman'],

'font.size': 12,

'axes.labelsize': 14,

'axes.titlesize': 16,

'xtick.labelsize': 12,

'ytick.labelsize': 12,

'legend.fontsize': 12,

'lines.linewidth': 1.5,

'axes.linewidth': 1.2,

'grid.alpha': 0.3,

'savefig.dpi': 600,

'savefig.format': 'pdf',

'savefig.bbox': 'tight'

})

# ====================

# 数据读取

# ====================

def read\_data(file\_path):

df = pd.read\_excel(file\_path)

# 智能日期检测逻辑

if '日期' in df.columns:

# 检查是否已经是datetime类型

if not pd.api.types.is\_datetime64\_any\_dtype(df['日期']):

# 尝试自动转换日期格式

try:

df['日期'] = pd.to\_datetime(df['日期'])

except:

# 处理Excel数值日期

df['日期'] = pd.to\_datetime(

df['日期'].astype(int), unit='D', origin='1899-12-30')

return df

train\_df = read\_data('train.xlsx')

val\_df = read\_data('val.xlsx')

test\_df = read\_data('test.xlsx')

future\_df = read\_data('future\_predictions.xlsx')

# ====================

# 可视化函数

# ====================

def plot\_comparison(ax, df, title):

"""训练/验证/测试集对比图"""

ax.plot(df['真实值'], color='#2c7bb6', label='Actual')

ax.plot(df['预测值'], color='#d7191c', ls='--', label='Predicted')

# 专业级格式设置

ax.xaxis.set\_minor\_locator(AutoMinorLocator())

ax.yaxis.set\_minor\_locator(AutoMinorLocator())

ax.spines[['top', 'right']].set\_visible(False)

ax.set\_title(title, pad=20)

ax.set\_xlabel('Time Step', labelpad=10)

ax.set\_ylabel('Value', labelpad=10)

# 修改后的图例设置（右上角定位）

ax.legend(frameon=True,

loc='upper right',

bbox\_to\_anchor=(0.98, 0.98), # 微调位置避免遮挡

framealpha=0.9, # 增加浅色背景

edgecolor='none') # 去除边框线条

def plot\_future(ax, df):

"""未来预测可视化"""

ax.plot(df['日期'], df['预测货量'], color='#d7191c')

# 专业时间序列格式

ax.xaxis.set\_minor\_locator(AutoMinorLocator())

ax.yaxis.set\_minor\_locator(AutoMinorLocator())

ax.spines[['top', 'right']].set\_visible(False)

ax.set\_title('Future Predictions', pad=20)

ax.set\_xlabel('Date', labelpad=10)

ax.set\_ylabel('Predicted Value', labelpad=10)

plt.setp(ax.get\_xticklabels(), rotation=45, ha='right')

# ====================

# 创建画布

# ====================

fig = plt.figure(figsize=(12, 16), dpi=600)

gs = fig.add\_gridspec(4, 1, height\_ratios=[1, 1, 1, 1.2], hspace=0.8)

# ====================

# 绘制子图

# ====================

ax1 = fig.add\_subplot(gs[0])

plot\_comparison(ax1, train\_df, 'Training Set Performance')

ax2 = fig.add\_subplot(gs[1])

plot\_comparison(ax2, val\_df, 'Validation Set Performance')

ax3 = fig.add\_subplot(gs[2])

plot\_comparison(ax3, test\_df, 'Test Set Performance')

ax4 = fig.add\_subplot(gs[3])

plot\_future(ax4, future\_df)

# ====================

# 保存输出

# ====================

plt.savefig('professional\_visualization.pdf')

plt.close()

'''

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from matplotlib.ticker import AutoMinorLocator

from matplotlib import rcParams

# ====================

# 学术图表样式设置

# ====================

plt.style.use('seaborn-v0\_8-paper')

rcParams.update({

'font.family': 'serif',

'font.serif': ['Times New Roman'],

'font.size': 12,

'axes.labelsize': 14,

'axes.titlesize': 16,

'xtick.labelsize': 12,

'ytick.labelsize': 12,

'legend.fontsize': 12,

'lines.linewidth': 1.5,

'axes.linewidth': 1.2,

'grid.alpha': 0.3,

'savefig.dpi': 1200,

'savefig.format': 'png', # 改为PNG格式

'savefig.bbox': 'tight'

})

# ====================

# 数据读取（保持不变）

# ====================

def read\_data(file\_path):

df = pd.read\_excel(file\_path)

if '日期' in df.columns:

if not pd.api.types.is\_datetime64\_any\_dtype(df['日期']):

try:

df['日期'] = pd.to\_datetime(df['日期'])

except:

df['日期'] = pd.to\_datetime(

df['日期'].astype(int), unit='D', origin='1899-12-30')

return df

train\_df = read\_data('train.xlsx')

val\_df = read\_data('val.xlsx')

test\_df = read\_data('test.xlsx')

future\_df = read\_data('future\_predictions.xlsx')

# ====================

# 可视化函数（优化标签旋转）

# ====================

def plot\_comparison(ax, df, title):

ax.plot(df['真实值'], color='#2c7bb6', label='Actual')

ax.plot(df['预测值'], color='#d7191c', ls='--', label='Predicted')

ax.xaxis.set\_minor\_locator(AutoMinorLocator())

ax.yaxis.set\_minor\_locator(AutoMinorLocator())

ax.spines[['top', 'right']].set\_visible(False)

ax.set\_title(title, pad=15)

ax.set\_xlabel('Time Step', labelpad=10)

ax.set\_ylabel('Value', labelpad=10)

ax.legend(frameon=True,

loc='upper right',

bbox\_to\_anchor=(0.98, 0.95),

framealpha=0.9,

edgecolor='none')

def plot\_future(ax, df):

ax.plot(df['日期'], df['预测货量'], color='#d7191c')

ax.xaxis.set\_minor\_locator(AutoMinorLocator())

ax.yaxis.set\_minor\_locator(AutoMinorLocator())

ax.spines[['top', 'right']].set\_visible(False)

ax.set\_title('Future Predictions', pad=15)

ax.set\_xlabel('Date', labelpad=10)

ax.set\_ylabel('Predicted Value', labelpad=10)

plt.setp(ax.get\_xticklabels(), rotation=35, ha='right') # 调整旋转角度

# ====================

# 创建2x2布局画布

# ====================

fig = plt.figure(figsize=(16, 12), dpi=600) # 调整画布尺寸

gs = fig.add\_gridspec(2, 2, hspace=0.5, wspace=0.3) # 增加水平间距

# ====================

# 分配子图位置

# ====================

ax1 = fig.add\_subplot(gs[0, 0]) # 第一行第一列

ax2 = fig.add\_subplot(gs[0, 1]) # 第一行第二列

ax3 = fig.add\_subplot(gs[1, 0]) # 第二行第一列

ax4 = fig.add\_subplot(gs[1, 1]) # 第二行第二列

# ====================

# 绘制子图

# ====================

plot\_comparison(ax1, train\_df, 'Training Set Performance')

plot\_comparison(ax2, val\_df, 'Validation Set Performance')

plot\_comparison(ax3, test\_df, 'Test Set Performance')

plot\_future(ax4, future\_df)

# ====================

# 保存为PNG格式

# ====================

plt.savefig('professional\_visualization25.png')

plt.close()

'''

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import savgol\_filter # 导入科学计算库

from matplotlib.ticker import AutoMinorLocator

from matplotlib import rcParams

# ====================

# 学术图表样式设置

# ====================

plt.style.use('seaborn-v0\_8-paper')

rcParams.update({

'font.family': 'serif',

'font.serif': ['Times New Roman'],

'font.size': 12,

'axes.labelsize': 14,

'axes.titlesize': 16,

'xtick.labelsize': 12,

'ytick.labelsize': 12,

'legend.fontsize': 12,

'lines.linewidth': 1.5,

'axes.linewidth': 1.2,

'grid.alpha': 0.3,

'savefig.dpi': 600,

'savefig.format': 'png',

'savefig.bbox': 'tight'

})

# ====================

# 数据读取与平滑处理

# ====================

def read\_and\_process(file\_path):

df = pd.read\_excel(file\_path)

# Savitzky-Golay滤波器参数

window\_size = 15 # 滑动窗口大小（奇数）

poly\_order = 2 # 多项式阶数

if '日期' in df.columns:

try:

df['日期'] = pd.to\_datetime(df['日期'])

except:

df['日期'] = pd.to\_datetime(df['日期'].astype(int), unit='D', origin='1899-12-30')

# 执行降噪处理

if '真实值' in df.columns:

try:

df['真实值\_平滑'] = savgol\_filter(df['真实值'],

window\_size,

poly\_order)

except Exception as e:

print(f"平滑处理警告：{str(e)}，使用原始数据替代")

df['真实值\_平滑'] = df['真实值']

return df

# ====================

# 加载数据集

# ====================

train\_df = read\_and\_process('train.xlsx')

val\_df = read\_and\_process('val.xlsx')

test\_df = read\_and\_process('test.xlsx')

future\_df = read\_and\_process('future\_predictions.xlsx')

# ====================

# 可视化函数（优化版）

# ====================

def plot\_smoothed\_comparison(ax, df, title):

# 绘制原始数据（半透明）

ax.plot(df['真实值'], color='#2c7bb6', alpha=0.2, label='Original')

# 绘制平滑数据

ax.plot(df['真实值\_平滑'], color='#2c7bb6', label='Smoothed Actual')

# 绘制预测值

ax.plot(df['预测值'], color='#d7191c', ls='--', label='Predicted')

# 格式设置

ax.xaxis.set\_minor\_locator(AutoMinorLocator())

ax.yaxis.set\_minor\_locator(AutoMinorLocator())

ax.spines[['top', 'right']].set\_visible(False)

ax.set\_title(title, pad=15)

ax.set\_xlabel('Time Step', labelpad=10)

ax.set\_ylabel('Value', labelpad=10)

ax.legend(frameon=True,

loc='upper right',

bbox\_to\_anchor=(0.98, 0.95),

framealpha=0.9,

edgecolor='none')

# ====================

# 创建专业布局

# ====================

fig = plt.figure(figsize=(18, 12), dpi=600) # 增加画布宽度

gs = fig.add\_gridspec(2, 2, hspace=0.4, wspace=0.25)

# 分配子图

ax1 = fig.add\_subplot(gs[0, 0]) # 训练集

ax2 = fig.add\_subplot(gs[0, 1]) # 验证集

ax3 = fig.add\_subplot(gs[1, 0]) # 测试集

ax4 = fig.add\_subplot(gs[1, 1]) # 未来预测

# ====================

# 绘制子图

# ====================

plot\_smoothed\_comparison(ax1, train\_df, 'Training Set (Smoothed)')

plot\_smoothed\_comparison(ax2, val\_df, 'Validation Set (Smoothed)')

plot\_smoothed\_comparison(ax3, test\_df, 'Test Set (Smoothed)')

# 未来预测保持原可视化

ax4.plot(future\_df['日期'], future\_df['预测货量'], color='#d7191c')

ax4.xaxis.set\_minor\_locator(AutoMinorLocator())

ax4.yaxis.set\_minor\_locator(AutoMinorLocator())

ax4.spines[['top', 'right']].set\_visible(False)

ax4.set\_title('Future Predictions', pad=15)

ax4.set\_xlabel('Date', labelpad=10)

ax4.set\_ylabel('Predicted Value', labelpad=10)

plt.setp(ax4.get\_xticklabels(), rotation=35, ha='right')

# ====================

# 保存输出

# ====================

plt.savefig('smoothed\_comparison.png')

plt.close()

print("可视化已保存为 smoothed\_comparison.png")

'''

**附录2：多目标优化算法**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from deap import base, creator, tools

import random

def set\_academic\_style():

"""设置学术出版级绘图样式"""

plt.style.use('seaborn-v0\_8-whitegrid')

plt.rcParams.update({

'font.family': 'Times New Roman',

'font.size': 12,

'axes.titlesize': 14,

'axes.labelsize': 12,

'xtick.labelsize': 10,

'ytick.labelsize': 10,

'legend.fontsize': 10,

'figure.dpi': 300,

'figure.figsize': (8, 5),

'savefig.dpi': 300,

'savefig.format': 'pdf',

'savefig.bbox': 'tight',

'axes.linewidth': 0.8,

'grid.linewidth': 0.4,

'lines.linewidth': 1.5,

'lines.markersize': 6,

'patch.edgecolor': 'black',

'text.usetex': False,

})

sns.set\_palette("tab10")

# -----------------------------

# 数据读取与预处理

# -----------------------------

data\_file = '附件1：物流网络历史货量数据.xlsx'

df = pd.read\_excel(data\_file)

def preprocess\_data(df):

# 计算每对节点的平均货流量

avg\_flow = df.groupby(['场地1', '场地2'])['货量'].mean().reset\_index()

return avg\_flow

# 构造货流矩阵：以“场地1”为行，“场地2”为列，缺失值填充为 0

def build\_flow\_matrix(avg\_flow):

flow\_matrix\_df = avg\_flow.pivot(index='场地1', columns='场地2', values='货量')

flow\_matrix = flow\_matrix\_df.fillna(0).values

return flow\_matrix

# -----------------------------

# 指标计算函数

# -----------------------------

def calculate\_adjustment(before, after):

# 计算重新分配前后各路径货流差异的平均值

return np.mean(np.abs(before - after))

def calculate\_network\_balance(flow\_matrix):

# 计算网络负荷均衡，采用标准差衡量

return np.std(flow\_matrix)

# -----------------------------

# 目标函数

# -----------------------------

def objective\_function(x, flow\_matrix, max\_capacity):

# x 为一维数组，转换为与 flow\_matrix 同形状

x = np.array(x)

new\_flow = x.reshape(flow\_matrix.shape)

# 目标：使得各路径的利用率接近1（即 new\_flow / max\_capacity 接近1），同时追求负荷均衡

cost = np.sum((new\_flow / max\_capacity - 1) \*\* 2)

balance = np.std(new\_flow)

return cost + balance, cost, balance

def global\_fitness\_function(x, flow\_matrix, max\_capacity):

return objective\_function(x, flow\_matrix, max\_capacity)[0]

# -----------------------------

# 遗传算法 (GA)（记录收敛历史）

# -----------------------------

def genetic\_algorithm\_with\_history(flow\_matrix, max\_capacity, generations=100, population\_size=50, crossover\_prob=0.7,

mutation\_prob=0.2):

# 避免重复创建

if "FitnessMin" not in creator.\_\_dict\_\_:

creator.create("FitnessMin", base.Fitness, weights=(-1.0, -1.0, -1.0))

if "Individual" not in creator.\_\_dict\_\_:

creator.create("Individual", list, fitness=creator.FitnessMin)

toolbox = base.Toolbox()

toolbox.register("attr\_float", random.uniform, 0, max\_capacity)

toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr\_float, n=flow\_matrix.size)

toolbox.register("population", tools.initRepeat, list, toolbox.individual)

toolbox.register("evaluate", objective\_function, flow\_matrix=flow\_matrix, max\_capacity=max\_capacity)

toolbox.register("mate", tools.cxBlend, alpha=0.5)

toolbox.register("mutate", tools.mutGaussian, mu=0.0, sigma=0.1, indpb=0.2)

toolbox.register("select", tools.selTournament, tournsize=3)

pop = toolbox.population(n=population\_size)

history = [] # 存储每一代最优个体的目标值

for gen in range(generations):

offspring = toolbox.select(pop, len(pop))

offspring = list(map(toolbox.clone, offspring))

# 交叉

for child1, child2 in zip(offspring[::2], offspring[1::2]):

if random.random() < crossover\_prob:

toolbox.mate(child1, child2)

del child1.fitness.values

del child2.fitness.values

# 变异

for mutant in offspring:

if random.random() < mutation\_prob:

toolbox.mutate(mutant)

del mutant.fitness.values

# 评估新个体

invalid\_ind = [ind for ind in offspring if not ind.fitness.valid]

for ind in invalid\_ind:

ind.fitness.values = toolbox.evaluate(ind)

pop[:] = offspring

best = min(pop, key=lambda ind: ind.fitness.values[0])

history.append(best.fitness.values[0])

return pop, history

def get\_best\_individual(population):

return min(population, key=lambda ind: ind.fitness.values[0])

# -----------------------------

# 粒子群优化 (PSO) 自定义实现（记录收敛历史）

# -----------------------------

def pso\_optimizer\_with\_history(flow\_matrix, max\_capacity, num\_particles=30, max\_iter=100, w=0.7, c1=1.5, c2=1.5):

dim = flow\_matrix.size

# 初始化粒子位置与速度

lb = np.zeros(dim)

ub = np.ones(dim) \* max\_capacity

positions = np.random.uniform(lb, ub, (num\_particles, dim))

velocities = np.random.uniform(-1, 1, (num\_particles, dim))

personal\_best\_positions = positions.copy()

personal\_best\_values = np.array([global\_fitness\_function(pos, flow\_matrix, max\_capacity) for pos in positions])

global\_best\_index = np.argmin(personal\_best\_values)

global\_best\_position = personal\_best\_positions[global\_best\_index].copy()

global\_best\_value = personal\_best\_values[global\_best\_index]

history = [global\_best\_value]

for it in range(max\_iter):

for i in range(num\_particles):

# 更新速度

r1, r2 = np.random.rand(dim), np.random.rand(dim)

velocities[i] = (w \* velocities[i] +

c1 \* r1 \* (personal\_best\_positions[i] - positions[i]) +

c2 \* r2 \* (global\_best\_position - positions[i]))

# 更新位置，并确保在 [lb, ub] 范围内

positions[i] = positions[i] + velocities[i]

positions[i] = np.clip(positions[i], lb, ub)

# 更新个人最优

fit = global\_fitness\_function(positions[i], flow\_matrix, max\_capacity)

if fit < personal\_best\_values[i]:

personal\_best\_positions[i] = positions[i].copy()

personal\_best\_values[i] = fit

# 更新全局最优

current\_best\_index = np.argmin(personal\_best\_values)

if personal\_best\_values[current\_best\_index] < global\_best\_value:

global\_best\_value = personal\_best\_values[current\_best\_index]

global\_best\_position = personal\_best\_positions[current\_best\_index].copy()

history.append(global\_best\_value)

return global\_best\_position.reshape(flow\_matrix.shape), history

# -----------------------------

# 混合优化算法（GA + PSO）

# -----------------------------

def hybrid\_optimizer\_with\_history(flow\_matrix, max\_capacity):

ga\_population, ga\_history = genetic\_algorithm\_with\_history(flow\_matrix, max\_capacity)

best\_ga = get\_best\_individual(ga\_population)

initial\_solution = np.array(best\_ga)

# 用 PSO 进一步细化，记录PSO收敛历史

pso\_solution, pso\_history = pso\_optimizer\_with\_history(flow\_matrix, max\_capacity)

# 混合历史：先拼接 GA 的最后值，再接 PSO 的历史

hybrid\_history = ga\_history + pso\_history

return pso\_solution, hybrid\_history

# -----------------------------

# 可视化函数

# -----------------------------

def plot\_convergence(histories, labels):

set\_academic\_style()

fig, axes = plt.subplots(3, 1, figsize=(8, 12), sharex=True)

colors = sns.color\_palette("husl", 3)

# 每个算法单独绘制

for idx, (ax, hist, label, color) in enumerate(zip(axes, histories, labels, colors)):

# 平滑处理

window\_size = max(1, len(hist) // 50)

smoothed = np.convolve(hist, np.ones(window\_size) / window\_size, mode='valid')

# 专业曲线绘制

ax.plot(smoothed,

color=color,

linestyle='-',

linewidth=1.6,

alpha=0.9,

label=label)

# 子图标题设置

ax.set\_title(f"{label} Convergence",

fontsize=12,

pad=8,

fontweight='bold',

color='#2c3e50')

# 坐标轴设置

ax.set\_ylabel("Objective Value",

fontsize=10,

labelpad=6,

fontweight='bold')

ax.yaxis.set\_major\_formatter(plt.FormatStrFormatter('%.1e'))

# 网格和边框设置

ax.grid(True, which='major', linestyle='--', alpha=0.7)

ax.grid(True, which='minor', linestyle=':', alpha=0.4)

ax.spines['top'].set\_visible(False)

ax.spines['right'].set\_visible(False)

ax.tick\_params(axis='both', which='both', direction='in')

# 添加半透明数据区域

x = np.arange(len(smoothed))

ax.fill\_between(x, smoothed,

alpha=0.15,

color=color,

zorder=2)

# 优化刻度密度

ax.xaxis.set\_major\_locator(plt.MaxNLocator(5))

ax.yaxis.set\_major\_locator(plt.MaxNLocator(5))

# 公共设置

axes[-1].set\_xlabel("Iteration",

fontsize=10,

labelpad=8,

fontweight='bold')

# 调整布局

plt.tight\_layout()

plt.subplots\_adjust(hspace=0.35)

# 保存输出

plt.savefig('separated\_convergence.pdf',

bbox\_inches='tight',

pad\_inches=0.1)

plt.show()

def plot\_comparison(best\_values, labels):

set\_academic\_style()

plt.figure(figsize=(6, 5))

# 创建专业条形图

bars = plt.bar(labels, best\_values,

color=sns.color\_palette("pastel"),

edgecolor='black',

linewidth=0.8,

alpha=0.9)

# 添加数值标签

for bar in bars:

height = bar.get\_height()

plt.text(bar.get\_x() + bar.get\_width() / 2., height \* 1.02,

f'{height:.2f}',

ha='center', va='bottom',

fontsize=10)

plt.xticks(rotation=15, ha='right')

plt.ylabel("Objective Value", fontweight='bold')

plt.title("Algorithm Performance Comparison",

fontsize=14, pad=12, fontweight='bold')

# 优化布局

plt.tight\_layout()

plt.grid(True, axis='y', linestyle='--', alpha=0.6)

plt.gca().spines['top'].set\_visible(False)

plt.gca().spines['right'].set\_visible(False)

# 保存图表

plt.savefig('performance\_comparison.pdf')

plt.show()

# -----------------------------

# 主函数

# -----------------------------

def main():

# 数据预处理

avg\_flow = preprocess\_data(df)

flow\_matrix = build\_flow\_matrix(avg\_flow)

max\_capacity = np.max(flow\_matrix)

# 用于记录各算法的最优目标值和收敛历史

best\_obj\_values = []

convergence\_histories = []

alg\_labels = []

# 遗传算法（GA）

ga\_pop, ga\_history = genetic\_algorithm\_with\_history(flow\_matrix, max\_capacity)

best\_ga = get\_best\_individual(ga\_pop)

best\_ga\_array = np.array(best\_ga).reshape(flow\_matrix.shape)

ga\_obj = global\_fitness\_function(best\_ga\_array.flatten(), flow\_matrix, max\_capacity)

best\_obj\_values.append(ga\_obj)

convergence\_histories.append(ga\_history)

alg\_labels.append("GA")

# 粒子群优化（PSO）

pso\_sol, pso\_history = pso\_optimizer\_with\_history(flow\_matrix, max\_capacity)

pso\_obj = global\_fitness\_function(pso\_sol.flatten(), flow\_matrix, max\_capacity)

best\_obj\_values.append(pso\_obj)

convergence\_histories.append(pso\_history)

alg\_labels.append("PSO")

# 混合优化算法（GA + PSO）

hybrid\_sol, hybrid\_history = hybrid\_optimizer\_with\_history(flow\_matrix, max\_capacity)

hybrid\_obj = global\_fitness\_function(hybrid\_sol.flatten(), flow\_matrix, max\_capacity)

best\_obj\_values.append(hybrid\_obj)

convergence\_histories.append(hybrid\_history)

alg\_labels.append("Hybrid")

# 可视化收敛曲线

plot\_convergence(convergence\_histories, alg\_labels)

pd.DataFrame(convergence\_histories).to\_csv('lishi.csv')

# 可视化各算法最优目标值对比

plot\_comparison(best\_obj\_values, alg\_labels)

if \_\_name\_\_ == "\_\_main\_\_":

main()